

**PCT**WORLD INTELLECTUAL PROPERTY ORGANIZATION  
International Bureau

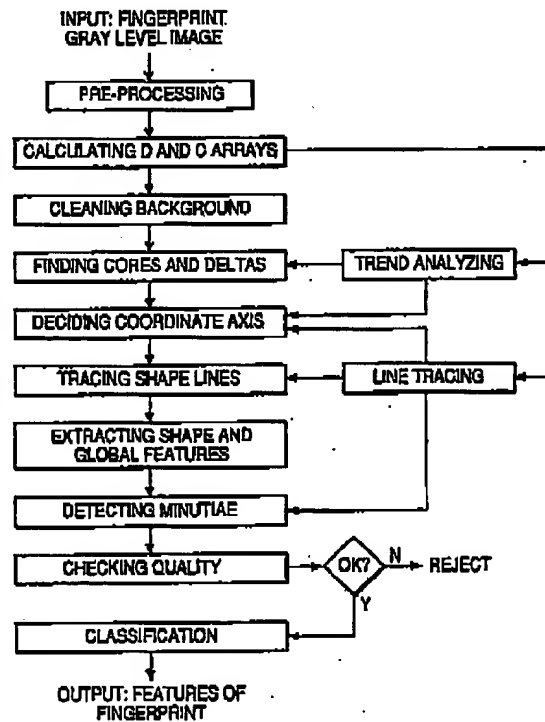
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>5</sup> : <b>G06K 9/00</b>		(11) International Publication Number: <b>WO 93/07584</b>
<b>A1</b>		(43) International Publication Date: 15 April 1993 (15.04.93)
(21) International Application Number: PCT/US92/08446		(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG).
(22) International Filing Date: 6 October 1992 (06.10.92)		
(30) Priority data: 772,393 7 October 1991 (07.10.91) US		
(71) Applicant: COGENT SYSTEMS, INC. (US/US); 3001-A West Mission Road, Alhambra, CA 91803 (US).		
(72) Inventor: SHEN, Xuening; 1500 1/2 Walnut Street, San Gabriel, CA 91776 (US).		
(74) Agents: FUNKELSTEIN, Jay, M. et al.; Spensly Horn Jubas & Lubitz, 1880 Century Park East, Fifth Floor, Los Angeles, CA 90067 (US).		Published With international search report.

(54) Title: METHOD AND SYSTEM FOR DETECTING FEATURES OF FINGERPRINT IN GRAY LEVEL IMAGE

## (57) Abstract

A method and automatic system (Fig. 1) for extracting both ordinary and unique features of a stripe pattern like a fingerprint from a gray level image without binary processing. A precise direction array (fig. 3) with the same number of image points as the image is generated by calculating the average direction of local ridges for every point in the image with a quick recurrent algorithm. There is also generated a curvature array in which each element presents the accuracy of the local average direction at corresponding points in the direction array and image. A region of clear ridges extracted from background and noise before detecting image features. The ridge trends (Figs. 8a-8e) and forkedness of a point are decided by analyzing the distribution of ridge directions on a circle around the point. For finding the cores and deltas, trend analysis is used for each singularity that is a maximum point on the curvature array. The coordinate axis that is consistent for various fingerprint types is decided by analyzing the structure of the direction array macroscopically.



Best Available Copy

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	MR	Mauritania
AU	Australia	GA	Gabon	MW	Malawi
BB	Barbados	GB	United Kingdom	NL	Netherlands
BE	Belgium	GN	Guinea	NO	Norway
BF	Burkina Faso	GR	Greece	NZ	New Zealand
BG	Bulgaria	HU	Hungary	PL	Poland
BJ	Benin	IE	Ireland	PT	Portugal
BR	Brazil	IT	Italy	RO	Romania
CA	Canada	JP	Japan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo	KR	Republic of Korea	SE	Sweden
CH	Switzerland	LI	Liechtenstein	SK	Slovak Republic
CI	Côte d'Ivoire	LK	Sri Lanka	SN	Senegal
CM	Cameroon	LU	Luxembourg	SU	Soviet Union
CS	Czechoslovakia	MC	Monaco	TD	Chad
CZ	Czech Republic	MG	Madagascar	TG	Togo
DE	Germany	ML	Mali	UA	Ukraine
DK	Denmark	MN	Mongolia	US	United States of America
ES	Spain			VN	Viet Nam
FI	Finland				

WO 93/07584

PCT/US92/08446

1

METHOD AND SYSTEM FOR DETECTING FEATURES OF FINGERPRINT  
IN GRAY LEVEL IMAGE

1. BACKGROUND OF THE INVENTION

1.1 Field Of The Invention

This invention relates to the automatic detection of both the common features (i.e. cores, deltas and minutiae) and the unique features (shape and global features) of a fingerprint by processing a gray level image of the fingerprint.

1.2 Description Of The Prior Art

The history of identifying and verifying individuality according to dermatoglyphic features is very long. Chinese people used to print their palms and fingers on documents and contracts as credit even since the seventh century A.D.. Although there are many other methods for identifying individuals today, fingerprint identification is still the most widespread and credible. However, since fingerprints became a legal identifier of persons about one hundred years ago, the number of fingerprint records has grown very quickly and manual management of the files has become very difficult. As a result, many automatic and semi-automatic systems for processing, recognizing, searching, and identifying fingerprints have been proposed.

The most widely used method for detecting features in many present automatic fingerprint identification systems is based upon binary image processing. A binary image is one in which each image element has one of only two binary values, e.g. 0 or 1. The key procedures in such processing are image enhancing, binarizing, thinning, smoothing and modifying. The minutiae of a fingerprint are detected by scanning the thinned binary image with a 3x3 window. Usually, the cores and deltas of fingerprints are also detected by scanning a binary image. A core exists where one or more lines of a fingerprint form a closed path, frequently a circle, or undergo an abrupt 180° direction change. A delta, as used herein, exists when three ridge lines

WO 93/07584

PCT/US92/08446

2

meet at a common point. A delta may be more accurately referred to as a Y.

In U.S. Pat. No. 4,083,035, an apparatus is provided for detecting the position (X and Y) and orientation angle ( $\theta$ ) of minutiae in a binary data bit stream of a 256x256 thinned image. The minutia orientation detector obtains an 8-bit vector average of all local angles present in each of a plurality of 8x8 bit windows across the image. This vector average of all of the local angles within a given 8x8 bit window is the orientation angle  $\theta$  for each minutia that is positioned within that given 8x8 bit window. There are 32x32 such windows on the image, i.e. a 32x32 ridge orientation array will be generated.

In U.S. Pat. No. 4,310,827, the minutia direction of an ending is defined as the direction of a single direction vector drawn from the ending to an arrival point, i.e. a skeleton point located by tracing a predetermined accurate length from the ending. The direction of a bifurcation is defined by a direction symmetrical to the average vector direction of three arrival points.

In U.S. Pat. No. 4,156,230, a 7x7 template scanning window is passed electronically over a 29x29 sub-array of the 32x32 ridge contour data as in U.S. Pat. No. 4,083,035 to generate a set of correlation values corresponding to each contour data element and to a plurality of reference angle vectors. The correlation values are processed for determination of peaks and valleys. The resultant data, representing the number of correlation peaks and the direction of each, provides 32 values which define the location and angular orientation of cores and deltas of a fingerprint.

In U.S. Pat. No. 4,151,512, the topological data, identifying singularity points such as tri-radii (i.e. deltas) and cores, as well as ridge flow line tracings related to those points, are extracted from a 32x32 ridge contour array as in U.S. Pat. No. 4,083,035. Subsequent to making the first cell tracing in any one direction from a tri-radius or core point, the information from the ridge contour array is used to supply additional angle data to continue each trace. Some logic

WO 93/07584

PCT/US92/08446

3

circuits determine the next row and column address incremental values according to a specification chart. The maximum length of a trace is 48 cells on the ridge contour array. Based upon the number of singularities located, an initial classification can be made wherein an arch is identified if no tri-radii are located, a whorl may be identified if two tri-radii are located and a general loop type may be identified if one tri-radius is located. The loop type pattern is classified according to the direction and size of the flow tracings by comparing them with a set of prestored references.

The following publications are also of relevance to the present invention:

Shen, "Several local properties of digital picture and their applications to the extraction of descriptive information of fingerprints", Acta Scientiarum Naturalium, Universitatis Pekinensis, No. 3, 1986, pp. 38-51. (In Chinese).

Shen, "The digital pseudo-curvature and its applications", Applied Mathematics, Sept. 1988, No. 3, Vol. 3, pp. 382-391. (In Chinese).

Shen et al., "A Similarity Measurement and Classification of Fingerprint", Proc. of 4th Chinese Conf. on Pattern Recognition and Machine Intelligence, 1984. (In Chinese).

### 1.3 Problems In The Prior Art

The problems listed here below relate to the manner of identifying, or designating, cores, deltas and the shapes of fingerprints based upon ridge directions in the patents cited above.

There are two problems in calculating ridge direction array: (1) Usage of the same value of ridge direction for every point in an 8x8 window will produce serious errors when the window is in a region where ridges curve significantly. (2) There is no measurement provided for representing the accuracy of each average ridge direction.

In U.S. Pat. No. 4,310,827, the direction of a minutia depend on the arrival points. So the direction will be effected

WO 93/07584

PCT/US92/08446

4

if any arrival point can not be found or if the skeleton ridges are not smooth enough.

There are four problems in analyzing ridge trends: (1) A 7x7 window in a ridge contour array, i.e. a 56x56 window in the original image, is too large to find cores and deltas of small whorls, or loops. (2) The fixed window size is not suitable for various types of cores and deltas. (3) The angular orientation with 32 values as well as cores and deltas with 29x29 positions are not accurate enough. (4) As many as 841 (=29x29) elements have to be analyzed for every fingerprint.

There are three problems in ridge flow tracing: (1) Each step in ridge flow tracing passes 8 points because every element in the ridge contour array refers to an 8x8 region in the image. This is too large for tracing at regions where ridges curve significantly. (2) The errors of position and direction are not accumulated to correct the trace. (3) The next step may be wrong when a core, delta or noise region is touched in tracing.

There are three problems in classification: (1) The initial classification based upon the number of deltas may be wrong in case an existing delta can not be found. (2) The loop sub-classification by comparing the rough flow tracing is sensitive to the initial fingerprint impression. (3) There is no sub-classification for whorls.

Finally, the main factor which effects the accuracy of fingerprint features extracted by binary processing is that much original information in a gray level image of the fingerprint may be lost after binarizing.

## 2. SUMMARY OF THE INVENTION

### 2.1 Objects Of The Invention

It is therefore a main object of the present invention to extract cores, deltas, minutiae, and shape and global features of a fingerprint from a gray level image based on original information as much as possible.

It is a more specific object of this invention to provide a quick algorithm for calculating the average direction of

WO 93/07584

PCT/US92/08446

5

local ridges at every point of a fingerprint and for generating a precise direction array.

Another object of the invention is to provide a measurement, termed local curvature, representing the accuracy of each local direction that is easy to calculate.

Another object of the invention is to provide a method for separating a region of clear ridges from background and noise in the image.

Another object of the invention is to provide a method for analyzing the ridge flow trends around any point in the image to decide its trend directions and forkedness.

Another object of the invention is to find the cores and deltas of a fingerprint by analyzing the trends only for each singularity of the image rather than analyzing the trends for every point of a direction array.

Another object of the invention is to locate the position of the center and central orientation of a plain arch of a fingerprint.

Another object of the invention is to establish a coordinate axis of any fingerprint that is consistent for various types and shapes of fingerprints.

Another object of the invention is to accurately trace shape lines, contour lines and normal lines of a fingerprint.

Another object of the invention is to classify fingerprints according to the structural relations among shape lines.

Another object of the invention is to extract shape features from the shape lines that are consistent for both whorls and loops, and to further classify fingerprints according to the shape features.

Another object of the invention is to extract global features of any fingerprint, including plain arch, however imperfect or partial it is and whatever type or shape it has.

Another object of the invention is to calculate the global difference between two fingerprints to finely classify and distinguish them.

WO 93/07584

PCT/US92/08446

6

Another object of the invention is to detect minutiae and their attributes from gray level images of fingerprints.

Another object of the invention is to calculate both the quality level and vector of fingerprints with regard to several aspects, for example noise level, area of clear region, position of center, number of minutiae, etc.

## 2.2 BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 is a processing flow diagram showing the sequence of basic steps characterizing the present invention.

FIG. 2 shows a point of an image and its neighborhood used to explain the calculation of local ridge direction and curvature according to the invention.

FIG. 3 shows a point and its four adjacent points for calculating four gradient models.

FIGS. 4a-4e show a symmetric convex region and its four subsets for calculating four average gradient models.

FIGS. 5a and 5b show the neighborhoods of two adjacent points and their common area as well as parts of the neighborhoods of two adjacent points and their common points.

FIGS. 6a-6d show various octagonal regions of clear ridges in a fingerprint image.

FIGS. 7a-7d show various types of fingerprint core patterns.

FIG. 7e shows a typical fingerprint delta pattern.

FIGS. 8a-8e show the ridge trends of various singularities and analysis circles.

FIGS. 9a-9e show the difference values within trend analyzing.

FIGS. 10a-10r show the shape lines of various fingerprints for 18 classes of shapes.

FIG. 11 shows the macroscopic structure of peripheral ridges of a fingerprint.

FIG. 12 shows a vault line and normal lines for locating the center of the coordinate axes of a fingerprint.

FIG. 13 shows a manner of determining the central orientation of a plain arch.

WO 93/07584

PCT/US92/08446

7

FIG. 14 illustrates the extraction of shape features extracted on shape lines.

FIG. 15 illustrates the extraction of global features of a fingerprint.

5 FIGS. 16a-16e show various minutiae.

FIGS. 17a-17d show the basic features of minutiae in terms of a ridge or valley respectively near a core or delta.

FIGS. 18a and 18b show two neighboring points in a tracing.

10 FIGS. 19-24 are pictorial views illustrating various tracing operations according to the invention on gray level fingerprint images, where FIGS. 19, 20 and 21 depict the tracing of lines for locating the center and center orientation of fingerprint patterns containing a whorl, a loop and an arch, respectively; and FIGS. 22, 23 and 24 depict the detection of  
15 global features for an arch, a loop and a whorl, respectively, according to the invention on the basis of the local directions of ridge lines of the fingerprint image patterns at points arranged along concentric circles.

20 FIG. 25 is a pictorial view illustrating extraction of minutiae from the region of an arch in a gray level fingerprint image.

### 2.3 GENERAL DEFINITIONS

25 There are many constants, parameters, variables and functions used herein. Some of these are defined in the C programming language as follows:

$x[]$  means an array named by  $x$ ; it can be defined as a set in which

30  $x[] = \{x[0], \dots, x[m-1]\},$

where  $m > 0$ .

$x[][]$  means a two dimensional array, or matrix; it can be defined as a set in which

$x[][] = \{x[0][0], \dots, x[m-1][n-1]\},$

35 where  $m > 0, n > 0$ . When each member of such a matrix is of the type  $x[i][j]$ , then, for each member,  $i$  is called the line, or row, number and  $j$  is called the column number.

WO 93/07584

PCT/US92/08446

8

$x = C ? y : z;$  means that if condition C is true, then  $x=y$ ; else  $x=z$ .

$\text{sign}(x) = (x < 0) ? -1 : 1;$

$\text{int}(x)$  means the largest integer that is not greater than

5  $x$ . Therefore, if  $x \geq 0$ , then  $\text{int}(x+0.5)$  means the most approximate integer of  $x$ .

$x \div y$  means the remainder, that is not smaller than 0, when  $x$  is divided by  $y$ .

10  $(x,y)$  means a digital point at coordinates  $x$  and  $y$  of a plane. It can also mean, in the appropriate context, a vector from origin point  $(0,0)$  to  $(x,y)$ .

$\text{dv}(X)$  means the direction of a vector  $X$ , which direction is in the range  $[0, 2\pi]$ .

$\#A$  means the cardinal or the number of members in set  $A$ .

15  $\Sigma(y(X), X, A)$  means the sum of values  $y(X)$  for every member  $X$  in set  $A$ .

$|x|$  means the model or absolute value of  $x$ . if  $x$  is a number then

$|x| = (x < 0) ? -x : x;$

20 if  $x=(x_1, \dots, x_n)$  is a vector, then

$|x| = \sqrt{\Sigma(x_i^2, i, (1, \dots, n))};$

$p_x$  means a parameter which may be predetermined or calculated for use in performing computations according to the invention. The range of each predetermined parameter will may  
25 be used in preferred embodiments of the invention will be listed below.

$\pi = 3.14159;$

$p_\pi = 252.$

### 30 DESCRIPTION OF THE PREFERRED EMBODIMENTS

#### 3.1 General Procedure

Referring to FIG.1, the general procedure of this invention for extracting as many features as possible of a fingerprint from a gray level image without binary processing  
35 is shown.

WO 93/07584

PCT/US92/08446

9

The significant steps in an exemplary method according to the invention are:

1) Input and Pre-processing

The input digital image has L rows and K columns of image elements, as shown in FIG.1. The intensity, or brightness level, in each image element has 3 to 1024, and preferably 256, gray levels and the image element density is 500 dpi (dots, or image elements, per inch) in each coordinate direction. For the further processing, the original range of gray levels of the fingerprint image is transformed into a uniform range.

2) Calculating Direction Array and Curvature Array

A direction array and a curvature array are calculated for each image point by a quick recurrent algorithm, possibly with the aid of some tables. The direction array and curvature array values for an image represent the average ridge direction and its accuracy or variance at each point of the image.

3) Cleaning Background

Segmenting an equiangular, or regular, octagonal region of clear ridges of a fingerprint from the background by eight straight lines according to the curvature array, and excavating all large conjunct regions of noise points in the octagon. All further processing is within this region.

4) Trend Analyzing

The ridge trends and forkedness of any point can be determined by analyzing the distribution of ridge directions around it on circles with different radii.

5) Finding Cores and Deltas

All singularities are found by scanning the curvature array with a 3x3 window to find all maximum curvature points. All cores and deltas are located by analyzing the trends for each singularity.

6) Line Tracing

Contour lines, shape lines and normal lines of a fingerprint are traced, based on the direction array,

WO 93/07584

PCT/US92/08446

10

accurately by accumulating the errors of coordinates and directions to correct the trace point by point.

#### 7) Deciding the Coordinate Axis

By macroscopically analyzing the structure of a direction array, the fingerprint is centered on a coordinate axis system; meanwhile the central orientation of the fingerprint is selected on the basis of the trend at the center.

#### 8) Classification

A fingerprint is classified into one of 18 classes according to the structural relations among the shape lines.

#### 9) Extracting Shape and Global Features

For classifying and fast searching, a few shape features are extracted consistently for various fingerprint classes except plain arch from shape lines. Furthermore a plurality of global features are extracted consistently for various fingerprint classes with reference to the coordinate axes.

#### 10) Detecting Minutiae

Based on the description of minutiae when the shape lines are valleys, all minutiae are detected by tracing each valley of the fingerprint in the gray level image. Each minutia is represented by its x, y coordinates and direction  $\theta$ .

#### 11) Quality Checking

A synthetic quality level and quality vector of fingerprints is presented based on the position of the center, number of minutiae, noise level, area of clear region, etc. in order to decide automatically, or to suggest to the operator, whether to accept, reject or reevaluate the fingerprint.

### 3.2 Direction Array and Curvature Array

All features of the fingerprint, including ordinary features (cores, deltas, minutiae) or novel features (shape and global features) are referenced to the average direction of ridges in a small region of the fingerprint. The calculation of local direction is very important for image processing and feature extraction of fingerprints. In this invention, an array whose every element represents an average direction of the

WO 93/07584

PCT/US92/08446

11

textures in a small region of the image is called a direction array.

There are three features of the method provided herein for calculating a precise direction array:

5 First, the direction array is calculated directly from the gray level image of the fingerprint, so the original information will be used as much as possible.

10 Second, the direction array is calculated point by point in the image, i.e. every element in the array is a local average direction of just one point in the image that is calculated on the basis of a neighborhood of the point. this is necessary spatially for regions on the fingerprint where the directions of ridges change greatly or the curvatures are very high.

15 Third, to represent the accuracy of the local direction at each point, a local average curvature of the textures in the neighborhood of the same point is also calculated, and a curvature array of the image is generated therefrom. The usage of local direction at a point should refer  
20 to the local curvature as the relation between an average value and its variance. A local curvature in some regions of a fingerprint, for example at a core, delta, scar or noise, will have very high curvature values which signify that local direction is meaningless there due to inconsistency of ridge  
25 directions and/or indistinct textures in that region. Generally, lower curvature values mean better accuracy of the direction value at a point.

For a given image area  $S$ , which may include the entire fingerprint image or any selected portion thereof, there are  
30 derived four summation gradient values  $v_i$  ( $i=1, 2, 3, 4$ ), each representing the sum of the absolute values of the difference,  $g_i(X)$ , in gray scale image point values,  $f(X)$ , between each pair of points,  $X$  and  $X-Q_i$ , in  $S$  for which  $Q_i$  is a selected vector. Thus, as will be seen from the example to be described,  
35 depending on the value of  $Q_i$ ,  $v_i$  is representative of the degree of change in the image across the image area in direction  $Q_i$ . Moreover,  $v_i$  will be particularly relevant to the point  $P$  at

WO 93/07584

PCI/US92/08446

12

the center of  $S$  because there is the greatest probability that it is the gradient condition at  $P$  which is described by  $v_i$ .

Table 1, below, provides an exemplary gray scale value matrix representing the gray values  $f(X)$  at respective points,  $X$ , of an image area  $S$ . Here, each point  $X$  has a horizontal coordinate  $k$  and a vertical coordinate  $l$ . As is apparent from Table 1, the origin of the coordinate system will be somewhere above and to the left of the illustrated image area.

Each point  $X$  is represented by a pair of coordinates  $k, l$ . In Table 1, the  $k$  and  $l$  coordinates at the center of  $S$  are  $n$  and  $m$ .

Table 1

	S	k\l	n-5	n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
			l										
15		m-5	13	14	15	15	15	14	12	11	9	7	6
		m-4	14	14	14	13	12	10	6	4	2	2	3
		m-3	13	11	9	6	6	4	3	3	4	6	8
		m-2	10	7	4	1	2	2	3	4	6	9	11
20		m-1	5	4	3	0	2	4	6	8	11	13	14
		m	3	4	5	6	8	9	11	12	14	15	15
		m+1	4	6	9	10	12	14	16	15	15	14	12
		m+2	7	9	12	13	14	14	15	14	13	10	8
25		m+3	13	14	14	14	14	13	10	8	8	6	4
		m+4	15	15	14	13	11	8	3	2	5	5	5
		m+5	14	12	10	8	7	4	0	1	5	7	8

For each vector  $Q_i$  there is produced a set  $S_i$  containing all points  $X$  for calculating  $g_i(X) = |f(X) - f(X - Q_i)|$ . The points in each set  $S_i$  are obtained by using the corresponding values for  $Q_i$ , i.e.,  $S_1$  is obtained by using  $Q_1$ ,  $S_2$  by using  $Q_2$ , etc.. The values for  $g_i(X)$  in each set  $S_i$  are given in Tables 2-5, below, where  $Q_i$  have the following  $k, l$  coordinate values,

35  $Q_1 = (1, 0);$   
 $Q_2 = (1, 1);$   
 $Q_3 = (0, 1);$   
 $Q_4 = (-1, 1).$

WO 93/07584

PCT/US92/08446

13

Table 2

$S_1$	$k$	n-5	n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
1	1											
5	m-5		1	1	0	0	1	2	1	2	2	1
	m-4		0	0	1	1	2	4	2	2	0	1
	m-3		2	2	3	0	2	1	0	1	2	2
	m-2		3	3	3	1	0	1	1	2	3	2
	m-1		1	1	3	2	2	2	2	3	2	1
	m		1	1	1	2	1	2	1	2	1	0
10	m+1		2	3	1	2	2	1	0	0	1	2
	m+2		2	3	1	1	0	1	1	1	3	2
	m+3		1	0	0	0	1	3	2	0	2	2
	m+4		0	1	1	2	3	5	1	3	0	0
	m+5		2	2	2	1	3	4	1	4	2	1
15												

Table 3

$S_2$	$k$	n-5	n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
20	1											
	m-5											
	m-4		1	0	2	3	5	8	8	9	7	4
	m-3		3	5	8	7	8	7	3	0	4	6
	m-2		6	7	8	4	4	1	1	3	5	5
25	m-1		6	4	4	1	2	4	5	7	7	5
	m		1	1	3	8	7	7	6	6	4	2
	m+1		3	5	5	6	6	6	4	3	0	3
	m+2		5	6	4	4	2	1	1	2	5	6
	m+3		7	5	2	1	1	4	7	6	7	6
30	m+4		2	0	1	3	6	10	8	3	3	1
	m+5		3	5	6	6	7	8	2	3	2	3

WO 93/07584

PCI/US92/08446

14

Table 4

$S_3$	$k$	n-5	n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
1												
m-5												
5	m-4	1	0	1	2	3	4	6	7	7	5	3
	m-3	1	3	5	7	6	6	3	1	2	4	5
	m-2	3	4	5	5	4	2	0	1	2	3	3
	m-1	5	3	1	1	0	2	3	4	5	4	3
	m	2	0	2	6	6	5	5	4	3	2	1
10	m+1	1	2	4	4	4	5	4	3	1	1	3
	m+2	3	3	3	3	2	0	0	1	2	4	4
	m+3	6	5	2	1	0	1	5	6	5	4	4
	m+4	2	1	0	1	3	5	7	6	3	1	1
	m+5	1	3	4	5	4	4	3	1	0	2	3

15

Table 5

$S_4$	$k$	n-5	n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	n+5
1												
m-5												
20	m-4	0	1	1	2	2	2	5	5	5	4	
	m-3	1	3	4	6	4	2	1	1	2	3	
	m-2	1	2	2	5	2	1	0	0	0	1	
	m-1	2	0	2	2	0	1	2	2	2	2	
	m	1	1	5	4	4	3	3	1	1	1	
25	m+1	0	1	3	2	3	3	3	1	0	1	
	m+2	1	0	2	1	0	1	0	1	1	2	
	m+3	4	2	1	0	0	2	4	5	2	2	
	m+4	1	1	0	1	2	2	5	6	1	1	
	m+5	1	2	3	3	1	1	2	4	0	2	

30

The number of values for  $g_i(X)$  on each set  $S_i$  is less than the number of points in area  $S$  because each value for  $g_i(X)$  is calculated only for a pair of values of  $f(X)$  and  $f(X-Q_i)$  which are both in area  $S$ . Thus, for example, on set  $S_1$  there will be no value of  $g_1(X)$  for which the  $k$  coordinate of  $P$  is  $n-5$  because the  $k$  coordinate of  $X-Q_1$  is  $n-6$ , which is outside of area  $S$ .

WO 93/07584

PCI/US92/08446

15

Then, for each set  $S_i$ , there is derived a value  $v_i$  where  $v_i = \Sigma g_i(P)$  for all values for  $g_i(P)$  on set  $S_i$ . Where  $i = 1, 2, 3, 4$ , the values for  $v_i$  in the example shown in Tables 1-5 will be  $v_1=167$ ;  $v_2=437$ ;  $v_3=337$  and  $v_4=194$ .

5 From these values for  $v_i$ , there will be derived four further values  $u_i$ , as follows:

$$\begin{aligned} u_1 &= \max(v_1, v_3) \\ u_2 &= \max(v_2, v_4); \\ 10 \quad u_3 &= \min(v_1, v_3); \\ u_4 &= \min(v_2, v_4). \end{aligned}$$

In the case of the example in Tables 1-5,  $u_1=337$ ;  $u_2=437$ ;  $u_3=167$ ;  $u_4=194$ .

15

These values are then used to derive:

$$e = \frac{1}{2}[u_3 + u_4 - \sqrt{(u_3 - u_4)^2 + 2(u_1 - u_2)(u_2 - u_4)}]$$

20 For the example shown in Tables 1-5,  $e=36.148865$ .

The derived values for  $v$ ,  $u$  and  $e$  are then used to derive a local direction value,  $d(f, P, S)$ , and a local curvature value,  $c(f, P, S)$ .

Referring to Figures 2, 3 and 4 of the accompanying  
25 DRAWING, according to the 1986 and 1988 publications of Shen, the formulas for calculating the local direction  $d$  and curvature  $c$  at each point  $P$  at the center of a set  $S$  in image  $f$  are as follows:

$$d(f, P, S) = \text{sign}(v_4 - v_2) \cdot \arctan((v_1 - e)/(v_3 - e)). \quad (1)$$

$$c(f, P, S) = \frac{|(u_2 + u_3) \cdot (u_2 + u_4) - 2u_1u_2|}{(u_1 + u_3) \cdot (u_2 + u_4) - 2u_3u_4} \quad (2)$$

30

WO 93/07584

PCT/US92/08446

16

In the example shown in Tables 1-5, which example is associated with  $P(n,m)$ ,  $d = -23.5^\circ$  and  $c = 0.0927$ .

The normal values of direction  $d$  are limited between  $-\pi/2$  (upward in FIG. 2) and  $+\pi/2$  (downward in FIG. 2). Here  $d=0$  represents the horizontal direction which points toward the right Table 1 and FIG. 2. According to Equation (2), the normal values of curvature  $c$  are all between 0 and 1. Specially,  $c=0$  represents the texture with a plain local curvature,  $c=1$  represents the texture with an abrupt local curvature. Additionally, for each background point or a noisy point  $P$ , both of  $c$  and  $d$  will be set to a special value that  $c(P)=255$  and  $d(P)=255$ .

$S(P)$  is a neighborhood of  $P$ , it is convex and symmetric with respect to  $P$  and with respect to the directions  $0^\circ$  and  $45^\circ$ , respectively. For example, a digital square, disc and octagon with  $P$  as center are all neighborhoods of this kind. Each  $S_i$  is a subset of  $S$  by deleting some border points as shown in FIG.4 where '.' is in  $S$  and  $S_i$ , '\*' is not in  $S_i$  ( $i=1,2,3,4$ ).

Now referring to FIGS.5, because there is tremendous complexity for calculating the direction array and curvature array per point, a quick recurrent algorithm with some tables is proposed to reduce the complexity on the basis of three key points:

First of all, because each gradient model  $|f(P)-f(P-Q_i)|$  will be used as many times as the number of the points whose neighborhood includes both  $P$  and  $P-Q_i$ , four arrays of gradient models at four directions as in FIG. 3 are calculated firstly and signed by  $g_i$  respectively:

$g_i(P) = |f(P) - f(P-Q_i)|$ , where  $i=1, 2, 3, 4$ . (3)

Second, there are many common points in the neighborhoods of both  $P$  and its adjacent point  $P^-$ . So  $v_i(P^-)$  can be calculated recurrently from  $v_i(P)$  by subtracting  $g_i(X)$  for each  $X$  at the left side  $L(S_i(P))$  of  $S_i(P)$  and adding  $g_i(X)$  for each  $X$  at the right side  $R(S_i(P^-))$  of  $S_i(P^-)$ , as shown in FIG. 5a where each '\*' is in  $S(P)$ , each '.' is in  $S(P^-)$  and each 'O' is in both sets:

WO 93/07584

PCT/US92/08446

17

$$v_i(f, P^-, S_i(P^-)) = v_i(f, P, S_i(P)) - \Sigma(g_i(X), X, S_i(P)) + \Sigma(g_i(X), X, R(S_i(P^-))). \quad (4)$$

For example, after  $v$  has been calculated for  $P(n, m)$ ,  $v_i$  may be calculated for  $P(n+1, m)$  by subtracting from  $P(n, m)$  the values of  $g_i(n-4, 1)$ , and adding the values of  $g_i(n+6, 1)$ , in each case 1 taking on each value from  $m-5$  to  $m+5$ .

Furthermore as shown in FIG. 5b, (where '\*' is in  $L$ , '.' is in  $L^\wedge$  and 'O' is in both of them,  $\Sigma(g_i(X), L^\wedge)$  also can be calculated recurrently:

$$\Sigma(g_i(X), X, L^\wedge) = \Sigma(g_i(X), X, L) - g_i(P') + g_i(P''). \quad (5)$$

Where  $P'$  is the top point in  $L$  and  $P''$  is the bottom point in  $L^\wedge$  shown in FIG. 5b.

Third, some look-up tables are used instead of a series of calculations for direction  $d$  and curvature  $c$ .

To calculate the value of  $d$ , a table

$$T_d = \{td[0], td[1], \dots, td[Md]\};$$

is created, where each term  $Td[k]$  in  $T_d$  is defined as

$$Td[k] = \arctan(k/Md), \quad \text{for } k = 0, 1, \dots, Md. \quad (6)$$

where  $Md$  is a predetermined integer. So the values of terms in  $T_d$  are all between 0 and  $\pi/4$ . For any group of  $v_i$  and  $e$  with the condition  $v_1 \leq v_3$ , let

$$\text{diff\_d} = Td[\text{int}(Md \cdot ((v_1 - e)/(v_3 - e)) + 0.5)] - \arctan((v_1 - e)/(v_3 - e)).$$

Then according to the continuity of the function  $\arctan()$ , the value  $|\text{diff\_d}|$  will be very small if  $Md$  is large enough. So that by the table  $T_d$ , when  $v_1 \leq v_3$ , Equation (1) can be transformed approximately to:

$$d(P) = \text{sign}(v_4 - v_2) \cdot ((v_1 \leq v_3) ? Td[\text{int}(Md \cdot ((v_1 - e) \cdot (v_3 - e)) + 0.5)] : (\pi/2 - Td[\text{int}(Md \cdot ((v_3 - e) \cdot (v_1 - e)) + 0.5])]). \quad (1')$$

The integer  $Md$  can be selected large enough to assure sufficient accuracy for  $y$ .

To calculate the value of  $c$ , another table

$$T_c = \{Tc[0][0], Tc[0][1], \dots, Tc[Mc][Mc]\};$$

is created where each term  $Tc[i][j]$  in  $T_c$  is defined as

WO 93/07584

PCT/US92/08446

18

$$Tc[i][j] = \frac{|(Mc+i)*(Mc+j) - 2(Mc)^2|}{(Mc+i)*(Mc+j) - 2*i*j},$$

where Mc is a predetermined integer. So the values of terms in Tc are all between 0 and 1. For any group of  $u_i$ , let

$$5 \quad \text{diff\_c} = Tc[\text{int}(Mc \cdot u_3/u_1 + 0.5)][\text{int}(Mc \cdot u_4/u_2 + 0.5)] - \frac{|u_1 + u_3| * (u_2 + u_4) - 2 * u_1 * u_2|}{(u_1 + u_3) * (u_2 + u_4) - 2 * u_3 * u_4}$$

Then according to the continuity of the function in Equation (2), the value |diff\_c| will be very small if Mc is large enough. So that by the table Tc, Equation (2) can be  
10 transformed approximately to:

$$c(P) = Tc[\text{int}(Mc \cdot u_3/u_1 + 0.5)][\text{int}(Mc \cdot u_4/u_2 + 0.5)].$$

The integer Mc can be selected large enough to assure sufficient accuracy of the formula.

The look-up tables may be provided, e.g. in a non-volatile  
15 addressable memory, where each entry corresponds to a respective value of Td or Tc.

If the neighborhood for calculating the local direction and curvature is a square window with  $2 \cdot r + 1$  points as both its length and width, let L and K represent the numbers of rows and  
20 columns, respectively, of an image array, and the algorithm is as follows:

```
<1> Calculate  $g_i$ 
       $g_1[l][k] = |f[l][k] - f[l][k-1]|$ ,      for  $l=0, \dots, L-1$ ,
25       $k=1, \dots, K-1$ ;
       $g_2[l][k] = |f[l][k] - f[l-1][k-1]|$ ,      for  $l=1, \dots, L-1$ ,
       $k=1, \dots, K-1$ ;
       $g_3[l][k] = |f[l][k] - f[l-1][k]|$ ,      for  $l=1, \dots, L-1$ ,
       $k=0, \dots, K-1$ ;
30       $g_4[l][k] = |f[l][k] - f[l-1][k+1]|$ ,      for  $l=1, \dots, L-1$ ,
       $k=0, \dots, K-2$ ;
```

WO 93/07584

PCI/US92/08446

19

```

        goto <2>;
<2> Accumulating  $g_i$  of each column  $k$  on a vertical line
    ((0,k), ... (2·r,k)).
         $w_1[k] = \sum (g_1[l][k]), 1, \{0,1,...,2·r\}, \text{ for } k=1,...,K-1;$ 
5        $w_2[k] = \sum (g_2[l][k]), 1, \{1,2,...,2·r\}, \text{ for } k=1,...,K-1;$ 
         $w_3[k] = \sum (g_3[l][k]), 1, \{1,2,...,2·r\}, \text{ for } k=1,...,K-1;$ 
         $w_4[k] = \sum (g_4[l][k]), 1, \{(1,2,...,2·r)\}, \text{ for } k=0,...,K-2;$ 
    where  $k$  is the number of a column,  $l$  is the number of a row
    and varies from 0 to 2·r, each  $w_i[k]$  is a summation of  $g_i$  for
10     $l=0,...,2·r$  and assigned by  $k$ .
        goto <3>;
<3> Accumulate  $w_i$  ( $i=1, 2, 3, 4$ ) in current window in each
 $g_i$  to obtain  $v_i$ .
         $v_1 = \sum (w_1[j]), j, \{1,...,2·r\};$ 
15        $v_2 = \sum (w_2[j]), j, \{1,...,2·r\};$ 
         $v_3 = \sum (w_3[j]), j, \{0,...,2·r\};$ 
         $v_4 = \sum (w_4[j]), j, \{0,...,2·r-1\};$ 
        goto <4>;
<4> Calculate curvature and direction of every point ( $k, l$ )
20 in current window and shifting the window to right.
        if ( $v_1+v_2+v_3+v_4 < p_v$ ), then
             $c[l][k] = 255; d[l][k] = 255;$ 
        else  $c[l][k] = Tc[\text{int}(Mc \cdot u_3/u_1 + .5)][\text{int}(Mc \cdot u_4/u_2 + .5)];$ 
            if ( $v_3 == v_1$ ), then  $d[l][k] = \text{sign}(v_4 - v_2) \cdot \pi/4;$ 
25         else ( $d[l][k] = \text{sign}(v_4 - v_2) \cdot ((v_1 > v_3) ?$ 
                 $Td[Md \cdot (v_1 - e)/(v_3 - e) + .5]:$ 
                 $(\pi/2 - Td[Md \cdot v_3 - 3] \cdot (v_1 - e) + .5))$ );
            )
        )
30      $k = k + 1;$ 
        if ( $k \geq K \cdot r$ ) goto <5>;
         $v_1 = v_1 + w_1[k+r] - w_1[k-r];$ 
         $v_2 = v_2 + w_2[k+r] - w_2[k-r];$ 
         $v_3 = v_3 + w_3[k+r] - w_3[k-r-1];$ 
35      $v_4 = v_4 + w_4[k+r-1] - w_4[k-r-1];$ 
        goto <4>;

```

WO 93/07584

PCT/US92/08446

20

<5> Put the current window at the beginning of the next line and recalculating  $w_i$ :

$l=l+1$ ;

if( $l \geq L-r$ ) then return;

5  $w_1[k] = w_1[k] + g_1[1+r][k] - g_1[1-r-1][k]$ , for  $k=1, \dots, K-1$ ;

$w_2[k] = w_2[k] + g_2[1+r][k] - g_2[1-r][k]$ , for  $k=1, \dots, K-1$ ;

$w_3[k] = w_3[k] + g_3[1+r][k] - g_3[1-r][k]$ , for  $k=0, \dots, K-1$ ;

$w_4[k] = w_4[k] + g_4[1+r][k] - g_4[1-r][k]$ , for  $k=0, \dots, K-2$ ;

$p\_v$  is a threshold which is directly proportional to  $r_2$ ,

10 and the condition  $v_1+v_2+v_3+v_4 < p\_v$  at a point means that the contrast of gray level in its neighborhood is very low.

Usually direction at those points are ignored, and the curvatures are assigned a special value of 255.

Here each term  $w_i[k]$  is associated with a column number

15 k. For example, referring again to the values in Tables 1 and 2 above, let  $r=5$ , then,

$$w_1[n] = \Sigma(g_1[1][n], 1, \{m-5, m-4, \dots, m+5\}) = 17,$$

similarly,  $w_1[n+1] = 26$ .

Now, an example is provided in Table 6, which is derived  
20 from Table 4, for  $r=2$ . The values of  $w_3[k]$  ( $k=n-5, n-4, \dots, n+5$ ) are calculated as in Table 6, while Table 7 shows the values of  $v_3$  at the points from line  $m-3$  to line  $m+3$  in Table 4.

WO 93/07584

PCT/US92/08446

21

Table 6

$w_3$	$k$	$n-5$	$n-4$	$n-3$	$n-2$	$n-1$	$n$	$n+1$	$n+2$	$n+3$	$n+4$	$n+5$
1												
5	m-3	10	10	12	15	13	14	12	13	16	16	14
	m-2	11	10	13	19	16	15	11	10	12	13	12
	m-1	11	9	12	16	14	14	12	12	11	10	10
	m	11	8	10	14	12	12	12	12	11	11	11
	m+1	12	10	11	14	12	11	14	14	11	11	12
10	m+2	12	11	9	9	9	11	16	16	11	10	12
	m+3	12	12	9	10	9	10	15	14	10	11	12

For example,

$$w_3[m+2][n] = w_3[m+1][n] - g_3[m-2][n] + g_3[m+2][n] = 14 - 2 + 0 = 12.$$

15

Table 7

$v_3$	$k$	$n-3$	$n-2$	$n-1$	$n$	$n+1$	$n+2$	$n+3$
1								
20	m-3	60	64	66	67	68	71	71
	m-2	69	73	74	71	64	61	58
	m-1	62	65	68	68	63	59	55
	m	55	56	60	62	59	58	57
	m+1	59	58	62	65	62	61	62
25	m+2	50	49	54	61	63	64	65
	m+3	52	50	53	58	58	60	62

For example,

$$v_3[m][n-1] = v_3[m][n-2] - w_3[m][n-4] + w_3[m][n+1] = 73 - 10 + 11 = 74.$$

30 So that with the above algorithm, only 4 additions and subtractions are needed on average for calculating  $v_3$  at each point.

### 3.3 Removing Background

35 For most fingerprint images, there always are noisy textures and other features in background. Before extracting the salient features of the fingerprint, it is necessary to

WO 93/07584

PCT/US92/08446

22

segment a region of clear ridges, or valleys, from background and noise, i.e. to decide the position and boundary of the region of clear ridges. The curvature value of a point in background is always very high due to the low contrast or noise, so the clear region can be obtained by cutting off the points with high curvature values.

Referring to FIGS. 6, the method proposed here is for locating an equiangular octagonal region of clear ridges of the fingerprint. The eight edges of the octagon are all straight lines with predetermined equispaced angular orientations, e.g. 0°, 45°, 90°, 135°, 180°, 225°, 270° and 315°, respectively.

Referring to Fig. 6a, the edges of an equiangular octagon are assigned  $E_1, \dots, E_8$  respectively, and the corners are assigned by the coordinates  $(x_1, y_1), \dots, (x_8, y_8)$  respectively. So each edge can be represented by an equation of straight line as follows,

$E_1: x+y=x_1+y_1; \text{ or } x+y=x_2+y_2;$   
 $E_2: y=y_2;$   
 $E_3: x-y=x_3-y_3; \text{ or } x-y=x_4-y_4;$   
 $E_4: x=x_4;$   
 $E_5: x+y=x_5+y_5; \text{ or } x+y=x_6+y_6;$   
 $E_6: y=y_6;$   
 $E_7: x-y=x_7-y_7; \text{ or } x-y=x_8-y_8;$   
 $E_8: x=x_8.$

Any equiangular octagon is determined by such eight equations, i.e. by eight parameters  $\{x_2, y_2, x_4, y_4, x_6, y_6, x_8, y_8\}$ . The octagon is obtained by cutting the image with eight lines according to the curvature array sequentially and described by only 8 bytes of the positions of eight lines. There are several shapes of octagons shown in FIGS. 6. The algorithm to locate eight edges is as follows:

<1> Set the original edges of clear region such that  
 $x_4=r; x_8=L-r-1; y_6=r; y_2=K-r-1;$   
 Calculate the average curvature of every row and column in the current clear region, and store them in the two arrays  $ac1[]$  and  $ack[]$  respectively, i.e.

WO 93/07384

PCT/US92/08446

23

```

      acl[y]=Σ(c[y][x],x,{x6,x6+1,...,x8}),
                                for y=y6, y6+1, ..., y2;
      acl[x]=Σ(c[y][x],y,{y6,y6+1,...,y2}),
                                for x=x6, x6+1, ..., x8.
5      let n=0;
      <2> if (n>pn) then goto <5>;
      where pn is a predetermined limitation, i.e. the times for
      cutting the image is not more than pn.
      Let x0 being the number or coordinate of the row of the
10     current array with the minimum average curvature value, i.e.,
      acl[y0]=min(acl[y6], ..., acl[y2]);
      similarly let y0 satisfy,
      ack[x0]=min(ack[x6], ..., ack[x8]).
      if(ack[x0]/(x8-x6+1) ≥ acl[y0]/(y2-y6+1)) then
15     goto <3>;
      else goto <4>;
      <3> Cut the area by horizontal lines, i.e. determine edges
      E2 and E6. Let
      pc_y=pc_p·Σ(acl[y], y, {y6,...,y2})/(y2-y6+1);
20     l1=max{y | (acl[y] > pc_y) & (y≤y0) & (y≥y6)};
      l2=min{y | (acl[y] > pc_y) & (y≥y0) & (y≤y2)};
      i.e. l1 is the largest row number satisfying l1≤y0, l2≥y6 and
      acl[l1]>pc_y; and l2 is the smallest row number satisfying
      l2≥y0, l2≤y2 and acl[l2]>pc_y. Then recalculate y2, y6 and
25     ack[] in the current clear region as follows,
      y2=l2; y6=l1;
      ack[x]=Σ(c[y][x], y, {y6, y6+1,...,y2});
                                for x=x6, x6+1, ..., x8.
      let n=n+1;
30     goto <2>;
      where pc_p is a predetermined parameter for calculating the
      parameters pc_x and pc_y.
      <4> Cut area by vertical lines, i.e. determine edges E6 and
      E8. Let
35     pc_x=pc_p·Σ(ack[i], i, {x6,...,x8})/(x8-x6+1);
      k1=max{x | ack[x]>pc_x & x≤x0 & x≥x6};
      k2=min{x | ack[x]>pc_x & x≥x0 & x≤x8};

```

WO 93/07584

PCT/US92/08446

24

i.e.  $k_1$  is the largest row number satisfying  $k_1 \leq x_0$ ,  $k_1 \geq x_4$  and  $ack[k_1] > p\_c\_x$ ; and  $k_2$  is the smallest row number satisfying  $k_2 \geq x_0$ ,  $k_2 \leq x_8$  and  $ack[k_2] > p\_c\_x$ . Then recalculate  $x_8$ ,  $x_4$  and  $ack[]$  in the current clear region as follows,

```

5       $x_8 = k_2$ ;  $x_4 = k_1$ ;
       $ack[Y] = \Sigma(c[Y][x], x, \{x_4, x_4+1, \dots, x_8\})$ ,
                                     for  $y = y_6, y_6+1, \dots, u_2$ ;

      let  $n = n-1$ ;
      goto <2>;
10 <5> Cut area by hypotenuse lines, i.e. determine the edges
       $E_1, E_3, E_5$  and  $E_7$  of the octagon enclosing the clear region or
      determine the numbers  $x_2, y_4, x_6$  and  $y_8$ . Let
       $p\_c\_z = (p\_c\_x + p\_c\_y)/2$ ;
      then the four numbers can be calculated as following,
15       $x_2 = \min\{z \mid (ack[z] > p\_c\_z) \ \& \ (z \leq x_8) \ \& \ (z \geq (x_4+x_8)/2)\}$ ;
      where for  $z = x_8, x_8-1, \dots, (x_4+x_8)/2$ ,
       $ack[z] = \Sigma(c[l][k], (k, l), A_2(z))$ ;
      where  $A_2(z) = \{(y_2, z), (y_2-1, z+1), \dots, (y_2-x_8+z, x_8)\}$ .
       $x_6 = \max\{z \mid (ack[z] > p\_c\_z) \ \& \ (z \geq x_4) \ \& \ (z \leq (x_4+x_8)/2)\}$ ;
20      where for  $z = x_4, x_4+1, \dots, (x_4+x_8)/2$ ,
       $ack[z] = \Sigma(c[l][k], (k, l), A_6(z))$ ;
      where  $A_6(z) = \{(y_6, z), (y_6+1, z-1), \dots, (y_6-x_4+z, x_4)\}$ .
       $y_4 = \min\{z \mid (ack[z] > p\_c\_z) \ \& \ (z \leq y_2) \ \& \ (z \geq (y_2+y_6)/2)\}$ ;
      where for  $z = y_2, y_2-1, \dots, (y_2+y_6)/2$ ,
25       $ack[z] = \Sigma(c[l][k], (k, l), A_4(z))$ ;
      where  $A_4(z) = \{(y_2, z), (y_2-1, z-1), \dots, (y_2+x_4-z, x_4)\}$ .
       $y_8 = \max\{z \mid (ack[z] > p\_c\_z) \ \& \ (z \geq y_6) \ \& \ (z \leq (y_2+y_6)/2)\}$ ;
      where for  $z = y_6, y_6+1, \dots, (y_2+y_6)/2$ ,
       $ack[z] = \Sigma(c[l][k], (k, l), A_8(z))$ ;
30      and  $A_8(z) = \{(y_6, z), (y_6+1, z+1), \dots, (y_6+x_8-z, x_8)\}$ .
      <7> After the octagon is obtained, the curvatures of points
      in the background are set to 255 for distinguishing them from
      the points in clear area.

```

### 35 3.4 Locating Singularities And Analyzing Trends

Now referring to FIGS.7, there are three types of core in a fingerprint, according to the structure of the ridges

WO 93/07584

PCT/US92/08446

25

around them, named 'o' (FIG. 7a), 'n' (FIG. 7b), and 'u' (FIG. 7c), respectively. An 'o' core may appear in a whorl, an 'n' may appear in a whorl, a double loop or a loop, while a 'u' may appear in a whorl, a double loop or a nodding loop.

5 However, any core or delta is a point such that the directions of ridges around it are very inconsistent, so its curvature is very high and may be higher than the curvatures of its neighboring points. Visually, there always are many bright points on the curvature array of a fingerprint; some

10 of them indicate the position of a core or delta, while others of them indicate a scar, fold or noise.

According to the publications of Shen in 1986, a point on a digital image is called a singularity if it has a maximum curvature value that is greater than a threshold

15 among its 8 neighboring points which form the corners and line midpoints of a square in which the point is centered.

A point:  $P=(k,l)$  is called a singularity if its curvature is not less than its 8 neighboring points' and not less than a predetermined threshold  $p\_c1$ , i.e.

- 20
1.  $(c(P) \geq c(P+Q_i)) \ \& \ (c(P) \geq c(P-Q_i)), \ (i=1,2,3,4);$
  2.  $c(P) \geq p\_c1;$

There will be some singularities appearing in the region near a core or delta. But usually there will be some singularities appearing in the region near a scar, fold or

25 noise, too. For the purpose of recognition of a core or a delta among singularities, analyzing the structure around a singularity is necessary.

Now referring to FIGS. 8, the ridge flow is different around a core, a delta or an ordinary point. The distinction

30 can be described easily by using the concept of ridge trend. A ridge trend of a point is defined as the direction of ridges that are near the point and run off from the point. There are three ridge trends for a delta, two trends for an 'a' core or ordinary point, one trend for a 'n' or 'u' core,

35 and no ridge trend for an 'o' core. The number of ridge trends of a point is called the forkedness of the point.

WO 93/07584

PCT/US92/08446

26

To find the ridge trends of a point  $P$ , a series of digital circles  $\{O_s\}$  with  $P_s$  as the center and various radii  $s$  are used. For every point  $P_s$  on a circle  $O_s$ , the difference between the local direction at  $P_s$  and the direction of vector  $P_c P_s$  is calculated and stored in an array  $dd[]$ . The difference when  $P_c P_s$  extends substantially in the direction of a ridge trend will be very small, as shown by the curve minima FIGS. 9, so the ridge trends will be decided by finding all minimums in  $dd[]$ . FIG. 9a shows the pattern of array  $dd[]$  around an 'o' core, FIG. 9b around an 'n' core, FIG. 9c around an 'a' core, FIG. 9d around a delta. Fig. 8a represents a noise point or a scar. Fig. 9e shows the pattern of  $dd[]$  around the point in Fig. 8e.

Fourier transform and reverse transform are used on  $dd[]$  for reducing the effect of noise. The forkedness can be decided by the power spectrum, while the trends can be found on filtered  $dd[]$ .

The method for determining the forkedness of a point will be described with reference to the power spectrum of a Fourier transform  $FT: \omega[0], \dots, \omega[j]$ ; where  $j$  is the order of Fourier the transform. Firstly, if  $\omega[0]$  is the maximum of all  $\omega[j]$  and  $\omega[0]$  is not smaller than a predetermined threshold  $p_{\omega_0}$ , then it is an 'o' core. Secondly, if  $\omega[1]$  is the maximum of all  $\omega[j]$  except  $\omega[0]$ , and  $\omega[1] \geq p_{\omega_1}$ , then it is a 'n' or 'u' core. Else if  $\omega[2]$  is the maximum of all  $\omega[j]$  except  $\omega[0]$  and  $\omega[2] \geq p_{\omega_2}$ , then it is an 'a' core. Else if  $\omega[3]$  is the maximum except  $\omega[0]$  and  $\omega[3] \geq p_{\omega_3}$ , then it is a delta point. Here  $p_{\omega_0}$ ,  $p_{\omega_1}$ ,  $p_{\omega_2}$  and  $p_{\omega_3}$  are all predetermined parameters.

In the case of Fig. 8e,  $\omega[2]$  to be the maximum, but it is not an 'a' core. To determine whether it is an 'a' core is depended on its trends. If the difference of the two trends is very large, for example, it is great than  $2\pi/3$ , then it would not be an 'a' core and would be ignored.

However it is notable that both the trend directions and forkedness at a point are due to the radius of the analysis

WO 93/07584

PCT/US92/08446

27

circle. The algorithm for analyzing the trends of a point P is as follows:

<1> Let  $s = s_{\min}$ .

Where  $s_{\min}$  is the minimum radius of digital circles for trend analysis and  $s_{\max}$  is the maximum.

<2> Calculate  $dd[]$ .

if ( $s > s_{\max}$ ) then reject and return;

else {for every point  $P_{si}$  on digital circle  $O_s$  do:

$dd[i] = |d(P_{si}) - dv(P_{si}-P)| \ \% \ \pi;$

if ( $dd[i] > \pi/2$ ) then  $dd[i] = \pi - dd[i];$

$nz = \#(X \mid (X \text{ on } O_s) \ \& \ (c(X) > p_{c2}));$

if ( $nz > p_{n2}$ ), then ( $s = s+1$ ; goto <2>)

else goto <3>;

i.e. if the number of points,  $P_{si}$ , on  $O_s$  with  $c(P_{si}) > p_{c2}$  is more than  $p_{n2}$ , then increase the radii  $s$  and recalculate  $dd[]$ .

<3> Derive Fourier transform on array  $dd[]$ .

$a[j] = \sum (dd[i] \cdot \cos(2 \cdot \pi \cdot i \cdot j / m), i, \{0, \dots, m-1\}) / m;$

$b[j] = \sum (dd[i] \cdot \sin(2 \cdot \pi \cdot i \cdot j / m), i, \{0, \dots, m-1\}) / m;$

for  $j = 0, \dots, n;$

where  $m = \#O_s$ .

The power spectrum is

$s = s_{\min} - 1; \ \omega[j] = (a[j])^2 + (b[j])^2; \ \text{for } j=0, \dots, n.$

where  $n$  is the order of the Fourier transform.

if ( $\omega[0] > p_{w0}$ ) {P is an 'o' core, return;}

else {let  $k$  satisfying,

$\omega[k] = \max\{\omega[1], \omega[2], \omega[3]\};$

if ( $\omega[k] < p_{\omega k}$ ) then goto <2>;

else if  $k$  is equal to 1 then

{calculating the trend direction,

$td = \arctan(b[1]/a[1]) \quad (-\pi < td < \pi);$

if ( $td > 0$ ) then P is an 'n' core;

else P is a 'u' core;

return;

}

else make the reverse Fourier transform:

$dd[i] = \sum (a[j] \cdot \cos(2 \cdot \pi \cdot i \cdot j / m) -$

WO 93/07584

PCT/US92/08446

28

```

        b[j].sin(2·π·i·j/m), j, {0,1,...,n});
    }
}
where  $p_{\omega_j}$  ( $j=0, \dots, 3$ ) are predetermined parameters.
5 <4> Finding each minimum value  $dd[i,j]$  in  $dd[i,]$ , i.e. for
each  $dd[i,j]$  that satisfies
 $dd[i,j] \leq dd[(i-1)\%m]$  &  $dd[i,j] \leq dd[(i+1)\%m]$  &  $dd[i,j] < p_{dd}$ ,
for  $j=1, \dots, l$ .
if (1 is not equal to k) then goto <2>;
10 else (the forkedness at P is k;
the trends at P are  $dv(P_{sij}-P)$ ,  $j=1, \dots, k$ .
)
return;
where  $p_{dd}$  is a predetermined threshold for finding minimum
15 points.  $dd[(i-1)\%m]$  and  $dd[(i+1)\%m]$  are the values of  $dd[i]$ 
at neighboring points of  $P_{sij}$  on the circle.
In the equations presented above,  $n$  is a constant that
is much smaller than  $m$ , the parameters  $s_{min}$  and  $s_{max}$  are
predetermined;  $p_c$ ,  $p_{n_2}$  and  $\omega_j$  ( $j=0, \dots, 3$ ) are thresholds.
20 More specifically,  $p_{n_2}$  is selected to eliminate curvature
values which are so large in number that the associated
direction value is unreliable. The threshold  $p_c$  is selected
to obtain points with a suitably high curvature value.
According to preferred embodiments of the invention,  $s_{min}$ 
25 may be equal to 5 and  $s_{max}$  may be equal to 20.
The procedure for finding all cores and deltas of a
fingerprint except the 'a' core is as follows:
<1> All singularities i.e. maximums of curvature array in
the clear region, are located;
30 <2> Every singularity is analyzed for finding trends by the
above algorithm.
<3> Select one point in every set composed of the same kind
of cores or deltas located together as the representative.
The criterion for selecting 'o' core is  $\omega[0]$ , for 'n' or 'u'
35 core is  $\omega[1]$ , for delta is  $\omega[3]$ .
In section 3.7, a method will be provided for finding
the 'a' core.

```

WO 93/07584

PCT/US92/08446

29

### 3.5 Line Tracing

A digital curve in a fingerprint is called a contour line if it is in keeping with the local ridge direction at every point on it. Usually, a contour line can be obtained by starting from a point on the fingerprint with a trend of the point as the initial direction and extending or tracing progressively according to the direction array. Especially, if the initial point is a core or delta of the fingerprint, then the extended trace is called a shape line of the fingerprint.

In the method provided below for accurately tracing contour lines, every tracing step moves by just one point; meanwhile the errors of each coordinate and direction are all accumulated for correcting the tracing; furthermore the tracing will stop at the right place. Where  $k_0$  and  $l_0$  are the column and row with minimum curvature in the clear region produced as described previously,  $cl$ ,  $ck$  and  $cd$  are the current row, column and direction values, the algorithm for tracing a contour line starting from point  $(k_0, l_0)$  with initial direction  $d_0$  is as follows:

$dl$ ,  $dk$ ,  $dd$  are the accumulating differences of coordinates and direction in tracing respectively;  $tll[]$ ,  $tlk[]$ ,  $tld[]$  are the arrays of coordinate and direction of referring points in tracing,  $di$  is a parameter,  $ac$  is the average curvatures in the segment of a line.

<1> Initializing the variables,

$dl=dk=ld=k_0$ ;  $ck=k_0$ ;  $cd=d_0$ ;

$i=1$ ;  $ac=0$ .

$tll[0]=l_0$ ;  $tlk[0]=k_0$ ;  $td[0]=d_0$ ;

goto <2>.

<2> Stepping to next point,

while  $|cd-tld[i-1]| \leq -\pi$   $cd = cd + \pi$ ;

while  $|cd-tld[i-1]| \geq \pi$   $cd = cd - \pi$ ;

if  $(|cd-d_0| > p_d)$  then goto <3>.

where  $p_d$  is the limitation of the difference between current direction  $cd$  and initial direction  $d_0$ .

$ac = ac + c[cl][ck]$ ;

WO 93/07584

PCT/US92/08446

30

```

      if (i ≥ p1) then
        {ac=ac-c[t11[i-di][t1k[i-di]]};
        if (ac > pac·p1) then goto <3>;
        where pac is a threshold.
5      dd = cd-tld[i-1];
      if (|dd| > pd2) then goto <3>;
      where pd2 is the limitation of accumulative difference of
      direction.
      if (dd > pd1) then
10        {dd > pd1;
          cd = cd-dd
          }
      else if (dd < -pd1) then
        {dd = dd+pd1;
15        cd = cd-dd;
          }
      else dd = 0;
      where pd1 means the maximum value for correcting the
      direction. The increase of cl and ck depend on sin(cd) and
20      cos(cd):
      if (|sin(cd)| < |cos(cd)|) then
        {cl = cl+sign(cos(cd)),
          dk = dk+tan(cd);
          if (|dk| ≥ 1) then
25            {ck = ck+sign(dk),
              dk = dk-sign(dk),
              }
          }
30      else then
        {ck = ck+sign(sin(cd)),
          dl = dl+ctan(cd);
          if (|dl| ≥ 1) then
            {cl = cl+sign(dl),
35            dl = dl-sign(dl),
              }
          }
    }

```

WO 93/07584

PCT/US92/08446

31

```

        if point (ck, cl) is out of the clear region, then goto
<3>;
        else, saving the current coordinates and direction,
            (tll[i]=cl, tlk[i]=ck, tld[i]=cd, i=i+1;
5         cd = d[cl][ck];
            goto <2>;
        }
<3>  Determining the length of the traced line,
        i = i-1,
10     if (c[tll[i]][tlk[i]] < p_ac) then goto <4>,
        else (i = i-1;
            if (i > 0) repeat <3>,
            else goto <4>;
        )
15  <4>  the traced line is {tll[j], tlk[j], tld[j], j=0,1,...,i;
        return,

```

where  $d_i$ ,  $p_{ac}$ ,  $p_{d_1}$ ,  $p_{d_2}$  and  $p_{d_3}$  are all predetermined parameters.

A contour tracing will be stopped if one or more of following conditions is true:

- (1) Current point is out of the clear region.
- (2) Average curvature of last several points is too high.
- (3) Rotated angle from initial direction is too large.

25 A similar algorithm is used to trace any normal line of a fingerprint; the normal line is defined as a digital curve on the image that is perpendicular to the local ridge direction everywhere. The algorithm can be obtained from the one above by replacing the assignment  $cd=d[cl][ck]$  in step

30 <2> with  $cd=d[cl][ck]+\pi/2$  or  $cd=d[cl][ck]-\pi/2$ . The normal lines are used in a novel method described in section 3.6, below, for macroscopically locating the center of a fingerprint.

35 Line tracing is a basic algorithm in this fingerprint processing system. It is important in locating the coordinate axes, extracting the shape features and detecting minutiae etc. that will be described in following sections.

WO 93/07584

PCT/US92/08446

32

The line tracing operation described in this section is used to trace shape lines from the center of a delta.

### 3.6 Macroscopic Method For Locating Coordinate Axes

5       The type or shape of a fingerprint, especially if it is characterized by a small whorl or loop, may be ambiguous due to distortion and noise produced when the impression is taken. Some impressions of a small whorl look like a loop or tent arch, while some impressions of a small loop look like a  
10       plain arch. Therefore, there must be some consistency in the rules for deciding the coordinate axes, i.e. the center and central orientation, of various types of fingerprints.

      In section 3.4, there was described a method for locating cores and deltas, except the 'a' core, of  
15       fingerprints by analyzing all singularities of the fingerprint. However, to find an 'a' core and to determine the center and central orientation of a plain arch that are consistent with other similar fingerprints, a method which involves analyzing the macroscopic structure of fingerprint  
20       ridges is needed.

      Referring to FIGS.10 again, many fingerprint types or shapes, for example whorls, double loops, loops, plain arch, tent arch etc., are shown. In fact, the principal  
25       distinctions among them are always at the central parts of the fingerprints, while the peripheries of fingerprints are all very similar. Generally, at the central area of any fingerprint, the ridges at the upper part will form a vault, the ridges at left and right sides will run off from the central part, and the ridges at the lower part will be always  
30       plain, as shown in FIG.11.

      Referring to FIG.12, any core of a fingerprint, except a 'u' core, is always at the most curved region of the ridges below the vault formed by upper ridges. Generally for any fingerprint except a nodding loop, the normal lines will,  
35       starting from the upper part and going down, all concentrate together at a central region where there are most curved ridges of the fingerprint, i.e. where there is an 'o' core or

WO 93/07584

PCT/US92/08446

33

an 'n' core for a whorl or loop, or an 'a' core for a plain arch. Some of the normal lines may end at the central region, while others may obviously curve at the central region, so the core can be located by analyzing the singularities near each end point and the most curved point on the normal lines. This method is very important for a plain arch, because it is usually difficult to determine, or locate, the center of a plain arch. The algorithm for locating macroscopically any core except a 'u' core of a fingerprint is:

10        Let l1 be the upper line border, l2 the lower line border, k1 the left column border and k2 the right border.  
       <1> Initializing, let the initial start point (k0, l0) for tracing be  
           l0 = l1; k0 = (k2-k1)/2;

15        <2> Selecting current start point (k0, l0),  
           l0 = l0+1;  
           if (c[l0][k0] > p\_c3 then goto <2>;  
           dk = p\_k·sin(d[l0][k0]);  
           if (|dk| > p\_d4) then {k0 = k0-dk; goto <2>;}  
 20        else, the start point is (k0, l0);  
           d0 = d[l0][k0];  
           goto <3>.

      where p\_c3 and p\_k are parameters.

      <3> Finding a vault line.

25        A vault line can be considered as a combination of two contour lines, i.e. a right contour line and a left contour line that both start from the middle of the print. So firstly the two contour lines should be traced. Starting from (l0,k0), two contour lines can be obtained by tracing  
 30        with directions d0 and d0+ $\pi$  respectively. These two contour lines are then combined into a vault line.

      If the vault is not perfect, i.e. if its length is too short or its chord is too slanting, or the curvature of the vault is too high, then goto <2>.

35        else let {vl[j], vk[j], vd[j], j=0,...,lv} be the vault line; where vl[j], vk[j], vd[j] are the y coordinate, the x

WO 93/07584

PCT/US92/08446

34

coordinate and the local direction, respectively, at the  $j$ th point on the line and  $lv$  is the length of the vault line.

let  $i=0$ ; goto <4>;

<4> Tracing a normal line by the previous algorithm with starting point  $(k0, l0)$  and direction  $d0$  that

$k0 = vk[i]$ ;  $l0 = vl[i]$ ;  $d0 = vd[i] + \pi/2$ ;

if  $(i \leq lv)$  then

( $i = i + p\_g$ ;

goto <4>;

10

)

else goto <5>;

where  $p\_d4$  is a threshold for limiting the local direction of starting point,  $p\_g$  is the gap between two normal lines at starting points.

<5> Determine the areas of concentration of the normal lines (see FIG. 12), then analyze the singularities in the area to find the 'o', 'n' and 'a' core or others by the forkedness with the algorithm described in 3.4.

In the above algorithm,  $p\_c$  is a threshold,  $p\_k$  is a constant. In the trend analyzing of above points, if the forkedness is 0 or 1, then the point is an 'o' or 'n' core, while if the number is 2, then it is an 'a' core.

By singularity analysis, there may be more than one 'a' core in the central region of the fingerprint. The criterion for selecting the most representative one among them is the angle difference  $dd$  between the two trends ( $d1$  and  $d2$ ) of a singularity, i.e.

$$dd = \min(|d2 - d1|, 2\pi - |d2 - d1|).$$

For example, if  $d1 = \pi/4$  and  $d2 = 3\pi/4$ , then  $dd = \pi/2$ ; if  $d1 = 2\pi$  and  $d2 = 3\pi/4$ , then  $dd = 3\pi/4$ .

The 'a' core which has the smallest angle difference will be selected as the most representative one.

Referring to FIG.13, there are two trends of an 'a' core, one is towards left while the other is towards the right. The main trend of an 'a' core is defined as the trend at the core side where the gap between two adjacent contour

WO 93/07584

PCT/US92/08446

35

lines is wider than at the other side. By this rule, the central orientation of a plain arch is consistent with loops.

After the trend analyzing for each singularity and locating the 'a' core macroscopically, the center and central orientation of a fingerprint can be decided as follows sequentially:

5 <1> If there is an 'o' core in the central region, then the pattern must be a whorl, the position of the center is the center of the core, the central orientation is  $\pi/2$ , i.e. down forward, and  $0^\circ$  is horizontal to the right.

10 <2> If, in the central region, there is an 'n' core and a 'u' core, the pattern is a whorl; if there is an 'n' core, no 'u' core and more than one delta, the pattern is a whorl; and if there is an 'n' core, no 'u' core and not more than one delta, the pattern is a loop. The position of the center is the same as the center of the core, and the central orientation is the trend of the core.

15 <3> Else if there is a 'u' core, then the pattern is a nodding loop, the position of the center is the same as the center of the core, and the central orientation is the trend of the core.

20 <4> Else if there is an 'a' core in the central region, then the pattern is a plain arch, the position of the center is the same as the center of the core, and the central orientation is the main trend of the core.

25 The center and central orientation of a whorl, a loop and an arch, decided by the macroscopic method are shown in FIGS. 19, 20 and 21, respectively. These FIGS. depict tracing lines which have been generated to be perpendicular to the local ridge directions in the vicinity of the center of the fingerprint pattern.

### 3.7 Shape Features And Classification

30 All of the shape lines of a fingerprint can be obtained by the above algorithm starting from the center of delta of the fingerprint with each trend as the initial direction. Line tracing is performed as described in Section 3.5, above.

WO 93/07584

PCI/US92/08446

36

Various shape lines of fingerprints, much as shown in FIGS.10, can serve to describe the shapes of fingerprints accurately. According to the structural relations of the shape lines, the fingerprints can be classified into 18 types each with a respective topological structure. A finer classification may be based on the shape features defined below.

FIG. 14 illustrates a technique for extracting the shape features of a left loop. Where  $C$  is the center,  $P_0$  is the delta,  $sl_1$ ,  $sl_2$  and  $sl_3$  are shape lines starting from  $P_0$ .  $P_1 \dots P_7$  are points selected on  $sl_2$  or  $sl_3$ . The algorithm for extracting the shape features of a loop is as follows:

- <1> Determine center  $C=(k_c, l_c)$  by the algorithm in Section 3.6;
- <2> Determine delta centered at  $P_0=(k_0, l_0)$  by the algorithm in Section 3.4;
- <3> Trace three shape lines  $sl_1$ ,  $sl_2$  and  $sl_3$  starting from  $P_0$  with three trends of the delta as initial directions respectively.
- <4> Selecting seven points  $P_i=(k_i, l_i)$ , ( $i=1, \dots, 7$ ) on  $sl_2$  and  $sl_3$ , such that

$$\angle v(P_i - C) = d_0 + i\pi/4,$$

where  $d_0 = \angle v(P_0 - C)$ . Because  $P_0$  is always very far from  $C$  or beyond the border of the image, no feature will be defined by referring to  $P_0$ .

<5> Calculate the distances between  $C$  and  $P_i$ , i.e.

$$|P_i - C| = \sqrt{(k_i - k_c)^2 + (l_i - l_c)^2}$$

<6> Ridge counting between  $C$  and  $P_0$ ,  
 Let  $\{(x_0, y_0), \dots, (x_n, y_n)\}$  be the straight line from  $C$  to  $P_0$ , where  $(x_0, y_0) = (k_c, l_c)$ ,  $(x_n, y_n) = (k_0, l_0)$ ; and let  
 $g[i] = f[y_i][x_i]$ , ( $i=0, \dots, n$ );  
 then the ridge counting is defined as,

WO 93/07584

PCT/US92/08446

37

$$rc(C, P_0) = \#(g[i] \mid g[i-1] > g[i] < g[i+1], (i=1, \dots, n-1)).$$

<7> Total of 18 shape features are defined, they are 7 distances  $|P_i - C|$ , one ridge count  $rc(C, P_0)$ , and 10 direction values referring to  $d_0$  including the central orientation, 3 trends of  $P_0$  and 6 local directions  $d(P_i)$ ,  $(i=1, \dots, 5, 7)$ .

The 18 shape features of a right loop are extracted in a manner similar to the left loop. A whorl or double loop can be considered as composed of two loops, i.e. one left loop formed by the left delta and center as well as one right loop formed by the right delta and center, so it has both 18 left loop shape features and 18 right loop shape features, i.e. 36 shape features in all. For a tent arch, the features referring to point  $P_0$  are not extracted because it may appear at either left or right. However, for the purpose of consistency between whorls and other shapes, a total of 36 shape features are supposed for any fingerprint. If some of the 36 shape features can not be obtained due to noise, imperfections in, or the shape of, the fingerprint, these features are each assigned a value of -1. There is no meaningful shape feature for a plain arch, in other words all shape features of a plain arch are equal to -1.

According to the structural relations of position and surrounding etc. among shape lines, fingerprints are classified into 18 classes with respectively different topological structure as shown in FIGS.10 which show 11 whorls, 4 loops, one accident, one tent arch and one plain arch. Every class of whorl, loop and tent arch can be further classified according to the shape features.

### 3.8 Global features and Global difference

The shape features for describing the pattern of a fingerprint are all based on both the center and delta. So they may be affected by the imperfections of fingerprint or by the noise which distorts the center, delta or shape lines. In particular, there is no shape feature defined for plain arch, so that for the purpose of practicality and consistency of fingerprint system, the features for describing the

WO 93/07584

PCI/US92/08446

38

pattern of a partial, noisy or plain arch fingerprint should be considered.

One of the most important parts in this invention is a method for defining and extracting the global features of various fingerprints to represent their pattern naturally and consistently by referring to the local ridge directions.

Generally, the global features of a fingerprint provide a basis method for representing the ridge direction array of the fingerprint. These features must be obtainable for any kind of fingerprint, and be effective in pattern matching of fingerprints.

The simplest method for defining global features is to select some points on the direction array and take the local ridge directions at each point as features. So if the amount of points is large enough, then the accuracy of repression will be fine enough. Especially, as shown in FIGS. 15, the points can be selected to form a circular, or polar, array or a rectangular array.

First method, the points can be selected on several circles with a common center. Referring to Fig. 15A,  $C=(k_0, l_0)$  is the center of a fingerprint,  $d_0$  is the central orientation, or direction. There are  $n$  circles  $O_i$  ( $i=0, \dots, n-1$ ), with a common center  $C$  and different radii  $r_i$  ( $i=0, \dots, n-1$ ). There are  $m_i$  selected points  $P_{ij}=(k_{ij}, l_{ij})$  ( $j=0, \dots, m_i-1$ ) on  $O_i$ , segmenting the circle equally ( $i=0, \dots, n-1$ ). The global features  $gf$  of a fingerprint are defined as a set:

$$gf = [gff((k_{ij}, l_{ij})) \mid j=0, \dots, m_i-1; i=1, \dots, n-1];$$

where

$$\begin{aligned} k_{ij} &= \text{int}(k_0 + r_i \cdot \cos(d_0 + j \cdot 2 \cdot \pi / m_i) + 0.5); \\ l_{ij} &= \text{int}(l_0 + r_i \cdot \sin(d_0 + j \cdot 2 \cdot \pi / m_i) + 0.5); \\ gff(P) &= \begin{cases} 255, & \text{if } c(P) > p\_c4; \\ \text{int}(d(P) \cdot p\_pi / \pi + 0.5) \cdot p\_pi, & \text{elsewhere.} \end{cases} \end{aligned}$$

i.e.,  $gf(P)$  is equal to 255 when the ridge directions around  $P$  are not clear or the curvature  $c(P)$  is greater than a predetermined threshold  $p\_c4$ ; else where  $gff(P)$  represents the direction  $d(P)$  in one byte with a value between 0 and

WO 93/07584

PCT/US92/08446

39

251; parameter  $p_\pi$  is selected to transform the range of angular value from  $[0, \pi]$  to  $[0, p_\pi]$  for storing it in one byte and reserving enough accuracy.

In an embodiment, the values of parameters are  $n=9$  and  $m_i=64$  ( $i=1, \dots, n$ ), so there are total  $9 \cdot 64=576$  points selected, and the global features of a fingerprint are composed by 576 bytes. If the number of points selected on each circle  $O_i$  is equal to others, then  $gf$  can be simply stored in an array of bytes:

$$gf=[gf[i][j] (=gff(P_{ij})) | j=0, \dots, m-1; i=0, \dots, n-1;]$$

10

For example, the global features of an arch, loop and whorl are shown in FIGS. 22, 23 and 24, respectively, where the center of each pattern is at the common center of the concentric circles and the central orientation of each pattern is represented by a short line extending from the center. There are 9 circles in each image, and each circle is composed of 32 points (for showing more clear than 64 points). For each point selected, if it is not in background and its curvature is not high, then the local direction is represented by a line centered on the selected point; the remaining selected points are each represented by a dot, as is particularly apparent at the bottom and lower portions of the left-hand and right-hand edges of FIG. 22.

The difference between two shapes of fingerprints always will reflect on their direction arrays. So that it also would reflect on the global features which represent the direction arrays. For this purpose, an important measurement, called global difference between two sets of global features, is necessary.

In the case of Equation (9), the global difference  $gd1$  between two fingerprints by their global features  $gf1$  and  $gf2$  is defined as:

$$gd1(gf1, gf2) = \min\{\sum(f_{dg}(d1-gf1[i][j], d2-gf2[i][(j+r)\%m])/\#M1(r), (i,j), M1(r)), r=0, \dots, m-1, \#M1(r)>0\}.$$

WO 93/07584

PCT/US92/08446

40

where  $d1$  and  $d2$  are central orientations of two fingerprints respectively, set  $M1(r)$  means

$$M1(r) = \{(i, j) \mid gf1[i][j] < p_\pi \text{ \& } gf2[i][(j+r)\%m] < p_\pi\};$$

function  $f\_dg()$  means

$$f\_dg(x, y) = f\_i(\min(|x-y|\%p_\pi, p_\pi - (|x-y|\%p_\pi)));$$

where  $f\_i(z)$  is an increase function of  $z$ . In the embodiment,  $f\_i(z) = z^2$ . So the global difference between two fingerprints is calculated by matching their global features with various radii  $r$  to find the minimum difference.

Second method, the points can be selected on a grid with  $n$  rows and  $m$  columns, as in FIG. 15b, so the global features can be stored in an array  $gf[][]$  such that:

$$gf[i][j] = d[y_{oo} + i \cdot dy][x_{oo} + j \cdot dx];$$

$$i = 0, \dots, n-1; \quad j = 0, \dots, m-1.$$

where  $x_{oo}$  and  $y_{oo}$  are the coordinates of the left upper corner point on the array,  $dy$  and  $dx$  are increases of row and column respectively.

The global difference  $gd2$  between two fingerprints by their global features  $gf1$  and  $gf2$  is defined as:

$$gd2(gf1, gf2) = \min(\sum(f\_dg(d1 - gf1[i]p[j], d2 - gf2[l][k]) / \#M2(l, k), (i, j), M2(l, k)),$$

$$\text{Complete } l = 0, \dots, n-1, k = 0, \dots, m-1.$$

The global difference can be used for finely classifying fingerprints in a database, or selecting similar fingerprints in a database to reduce the difficulty of minutia matching during a searching procedure.

### 3.9 Detecting Minutia From Gray Level Image

Minutiae are very important traditional features of a fingerprint, and are used in final verification of identity of two fingerprints. Usually minutiae are described with respect to the pattern of fingerprint ridges. There are many types of minutiae on a fingerprint, for example as shown in FIGS. 16, endings (a), bifurcations (b), islands (c), eyes (d), bridges (e), etc. In brief, minutiae are singularities of ridges.

However ridges always coexist with valleys on a fingerprint, and each feature or minutia of ridges always

WO 93/07584

PCT/US92/08446

41

corresponds to a change in valleys, so that minutiae can be described in terms of valleys, too. In general, an ending of a ridge is a bifurcation of valleys while a bifurcation of ridges is an ending of a valley, an island of a ridge is an eye of a valley and an eye of a ridge is an island of a valley. Referring to FIGS. 17, the exceptions may appear at cores and deltas. The description of minutiae that are just at a core or delta in terms of ridges is different from the description in terms of valleys. However, the descriptions of minutiae should be consistent by being all in terms of valleys or all in terms of ridges.

For automatic detection of minutiae, the novel method provided here is based upon tracing the valleys rather than the ordinary method which is based upon binarizing, thinning and smoothing the ridges. Generally in a fingerprint image, the quality of valleys is much better than that of ridges, primarily because of the following reasons: firstly, there are no sweat glands in valleys; secondly, the widths of valleys are more even than those of ridges; and thirdly, the gray levels in valleys are more even than in ridges. Although there will be incipient ridges in the valleys of some fingerprints that may affect valley tracing, all ridges of every fingerprint have sweat glands that may affect ridge tracing. So in general, the result of valley tracing should be much better than ridge tracing.

The algorithm for tracing a valley with an initial point  $(k_0, l_0)$  and direction  $d_0$  is similar to tracing a line, except that it uses a key technique that keeps the step points in the valley.

Let  $f[l][k]$  be an image array. Its element  $f[l][k]$  equals the gray scale value of a point  $(k, l)$ , it will be set to -1 after it has been traced.  $ag$  is the summation of gray scale values of the last  $p_l$  points in a tracing line. Array  $tlg[]$  is used to store gray scale values of traced points. The definitions of other variables are the same as for the algorithm of line tracing in Section 3.5.

<1> Initializing the variables.

WO 93/07584

PCI/US92/08446

42

```

      dl=dk=dr=0;
      cl=l0; ck=k0; cd=d0;
      i=1; ac=0.
      tll[0]=l0;      tlk[0]=k0;      tld[0]=d0;
5    tlg[0]=f[l0][k0];
      goto <2>.
<2>  Step to the next point,
      Accumulate curvatures of every point in the valley,
      ac=ac+c[cl][ck];
10   if (i>p_1) then
        {ac=ac-c[tll[i-p_1]][tlk[i-p_1]];
        if (ac > p_ac·p_1) then goto <3>;
        }
      Accumulate gray scale values of every point in the
15   valley.
      ag=ag+f[cl][ck];
      if (i > p_1) then
        {ag=ag-tlg[i-p_1];
        if (ag < p_ag·p_1) then goto <3>;
        }
20   if (cd-tld[i-1] ≤ -π) then cd = cd+π;
      if (cd-tld[i-1] ≥ π) then cd = cd-π;
      if (|cd-d0| > p_d3) then goto <3>,
      where p_d3 is the limitation for changing current direction
25   cd per step in tracing.
      ac=ac+c[cl][ck];
      if (i ≥ p_1) then
        {ac=ac-c[tll[i-p_1]][tlk[i-p_1]];
        if (ac > p_ac·p_1) then goto <3>;
        }
30   where p_ac is a threshold.
      }
      dd=cd-tld[i-1];
      if (|dd| > p_d2) then goto <3>;
      where p_d2 is the limitation of accumulative difference of
35   direction.
      if (dd > p_d1) then
        {dd = dd-p_d1;

```

WO 93/07584

PCT/US92/08446

43

```

        cl=cd-dd
    }
    else if (dd < -p_dl) then
        {dd=dd+p_dl;
5      cl=cd-dd;
        }
    else dd=0;
    where p_dl means the maximum value for correcting the
    direction. The increase of cl and ck depend on sin(cd) and
10  cos(cd);
        if (|sin(cd)| < |cos(cd)|) then
            {cl=cl+sign(cos(cd)),
              dl:=dk+tan(cd);
              if (|dk| ≥ 1) then
15              {ck=ck+sign(dk).
                  dk=dk-sign(dk),
                  }
              }
        if any point (x,y) of ((ck,cl), (kl,ll), (kr,lr)) is out
20  of the clear region or f[y][x]<0, then goto <3>;
        else save current coordinate, direction and gray level
        values;
            {tll[i]=cl; tlk[i]=ck; tld[i]=cd; i=i+1;
              f[cl][ck]=-1;
25      cd=d[cl][ck];
              dd=dd+(f[ll][kl]-f[lr][kr])*p_ga;
              goto <2>;
            }
    where (kl,ll) and (kr,lr) shown in FIGS. 18 are called the
30  left point and right point of current point (ck,cl)
    respectively. Both of them are the 4-neighboring points of
    current point and 8-neighboring points of the previous point
    (tlk[i],tll[i]). p_ga is a predetermined parameter of
    modifying direction by difference of gray scale values.
35  <3> Determine the length of traced valley,
        i=i-1;

```

WO 93/07584

PCT/US92/08446

44

```

        if ((c[tll[i]][tlk[i]]<p_ac) & tlg[i]>p_ag)) then goto
<4>,
        else {f[tll[i]][tlk[i]]=tlg[i];
              i=i-1;
5      if (i>0) then repeat <3>,
          else goto <4>.
        }
<4> The traced valley is      (tll[j], tlk[j], tld[j],
(j=0,1,...,i-1),
10      return;
      Where p_1 is a constant, p_ac, p_d1 and p_d2 are all
      thresholds.
      Referring to FIGS.18, both the left point (kl,1l) and
      the right point (kr,1r) are 4-neighboring for current point
15 (ck,cl) and 8-neighboring for previous point
      (tlk[i-1],tll[i-1]). (ck,cl) may be replaced by its
      4-neighboring point (kl,1l) or (kr,1r) according to their
      gray level. This algorithm is similar to that for line
      tracing with gray level as an additional factor.
20      In detail, the tracing line will firstly step from the
      prior point to the current point temporally, then the gray
      scale of two 4_neighboring points, i.e. (kl,1l) and (kr,1r),
      of the current point are considered. A point is selected to
      be a valley point if its gray scale value is higher than or
25 equal to the other points in the neighborhood.
      A valley tracing will be stopped if one or more of
      following conditions is true:
        (1) The current point is out of the clear region.
        (2) The average curvature ac of the last p_1 points in
30 the tracing is very high, i.e.greater than p_ac.
        (3) Any previous valley trace is touched.
        (4) The average gray level ag of the last p_1 points in
      the tracing is very low, i.e. less than p_ag.
      The algorithm for detecting minutiae from a gray level
35 image by valley tracing is as follows:
      <1> Let gp=cap2;

```

WO 93/07584

PCT/US92/08446

45

<2> The start point  $P$  for valley tracing should satisfy each of following conditions:

(1)  $P$  is a maximum point in a  $3 \times 3$  neighborhood on gray scale image  $F$  and  $f(P) > p_f$ .

5        A maximum point in a gray scale image means it is one whose gray scale value is not less than that of each of its 8 neighboring points.

(2)  $P$  is in the clear region of ridges and  $c(P) < p_c$ . This means the curvature at  $P$  is smaller than  $p_c$ .

10        (3) There is no traced line at directions  $c(P) + \pi/2$  and  $c(P) - \pi/2$  within distance  $g_p$ .

For every such point  $P$ , the valley is traced in two initial directions  $c(P)$  and  $c(P) + \pi$ , respectively. The minutiae are detected as a result of the conditions for  
15        stopping a trace: If the trace is stopped due to condition (3), then a valley bifurcation is found; While if the trace is stopped due to condition (4), then a valley ending has been found.

<3> Connect any two terminals of traces (a terminal being  
20        the start or end of a trace) if:

(1) The two last directions of the traces are opposite to one another;

(2) The positions of two terminals of traces are very close;

25        (3) The average gray level between them is higher than  $p_{ag}$ .

<4>  $g_p = g_p - 1$ ; if  $g_p > gap_1$ , then goto <2>;

Where parameters  $p_f$ ,  $t$ ,  $gap_1$  and  $gap_2$  are all experimentally determined constants. Each minutia found is  
30        described by three attributes, i.e.  $x$ ,  $y$  and  $\theta$ . The coordinates  $x$  and  $y$  are same as the position of a trace terminal, the direction  $\theta$  is equal to the one of  $d[y][x]$  and  $d[y][x] + \pi$  which is closer to the last direction in the trace.

An example of extracting minutiae by tracing valleys on  
35        a gray level of a fingerprint is shown in FIG. 25, where the fingerprint is the same as in FIGS. 21 and 22 and the gray level is reversed.

WO 93/07584

PCT/US92/08446

46

### 3.10 Quality Level And Vector

The features of a fingerprint may be effected by many factors, for example noise level, the effective area of the clear region, the position of the center, the number of minutiae, and so on. Sometimes the factors are due to the quality of the finger itself, while at other times they are due to the impression or input device. A quality level  $q_l$  should be provided after image processing in order to make possible an automated or operator controlled decision as to whether to accept, reject, or reinput the fingerprint image, or if possible to take a new fingerprint impression. The quality level can be described in detail by a quality vector:

$$q_v = (q_n, q_a, q_p, q_m, q_h);$$

where each factor is calculated as follows:

<1> The noise level  $q_n$  refers to the average curvature in the whole clear region, i.e.

$$q_n = f_{q_n}(a_c);$$

where

$$a_c = \Sigma(c(X), X, \{X | c(X) < 1\}) / \# \{X | c(X) < 1\};$$

$f_{q_n}(z)$  is an increase function for  $z$  in the range 0 to 1.

In the embodiment  $q_n$  is defined as,

	0,	when $a_c \leq c1$ ;
	1,	when $a_c > c1$ & $a_c \leq c2$ ;
$q_n$	2,	when $a_c > c2$ & $a_c \leq c3$ ;
	3,	when $a_c > c3$ .

Where  $c1$ ,  $c2$ , and  $c3$  are all predetermined experimental values. If the average curvature of an image is very small, then  $q_n = 0$ , i.e. the image's quality is good; while if the average curvature is large, then the noise in the image would affect the processing,  $q_n$  will equal to 1, 2 or 3 depending on the noise.

<2> The effective area  $q_a$  represents the number of global features which define a direction, in the case of Equation (9),

$$q_a = f_{q_a}(\# \{x | (x \text{ in } gf) \ \& \ (x < p_r)\});$$

where function  $f_{q_a}(z)$  is increasing for  $z$  between 0 and 255, to ensure that  $q_a$  is 1, 2 or 3. When  $q_a$  equals 0, the

WO 93/07584

PCT/US92/08446

47

quality of the image is good, otherwise the larger the value, the worse the quality.

<3>  $q_p$  depends on whether the position of the center  $(k_0, l_0)$  is in the central region CR of the image, i.e.

5  $q_p = (\text{is } (k_0, l_0) \text{ in CR}) ? 0 : 1;$

In the embodiment,

$CR = \{(k, l) \mid (L/4 < l < L \cdot 2/3) \ \& \ (K/3 < k < K \cdot 2/3)\}.$

<4>  $q_m$  depends on the number and average quality of minutiae  $a_{mq}$ , i.e.

10  $q_m = \begin{cases} 0, & \text{if } nm < p_{nm}; \\ f_{mq}(a_{mq}), & \text{otherwise,} \end{cases}$

where  $p_{nm}$  is a predetermined threshold,  $f_{mq}(z)$  is an increase function for  $z$ . In the embodiment,  $p_{nm}=18$ ,

15  $f_{mq}(z)=z/4$ .

<5>  $q_h$  means help level that presents the reliability of the center  $C$  by the average curvature around  $C$ ,

$q_h = f_{qh}(\Sigma(c(X), X, NC)/\#NC);$

where

20  $NC = \{X \mid |X-C| \geq p_{r1} \ \& \ |X-C| \leq p_{r2}\}.$

$f_{qh}(z)$  is an increase function for  $z$ .

Finally,

$q_l = q_n + q_a + q_p + q_m + q_h;$

It is anticipated that the invention will be implemented  
25 by means of a general purpose digital computer system which is programmed in accordance with the algorithms described above and is provided with an appropriate graphics input device capable of scanning a fingerprint image and inputting gray level image point brightness values and displaying and  
30 printing and writing the results of the image processing procedures in an output device.

In an embodiment of the above method, the following parameters may have values in the ranges specified below:

	Section	Parameter	Range
35	3.2	$r$	[5, 30]
		$p_v$	$[4 \cdot r \cdot r, 60 \cdot r \cdot r]$
	3.3	$p_{n1}$	[1, 5]
		$p_{c_p}$	[0.5, 1]

WO 93/07584

PCT/US92/08446

48

5	3.4	p_c1	[0.6, 1]
		p_w0	[1.5, 2.5]
		p_w1	[0.5, 1.5]
		p_w2	[0.5, 1.5]
		p_w3	[0.5, 1.5]
10	3.5	p_c2	[0.6, 1]
		p_n2	[1, 10]
		p_dd	[0.1, 0.6]
		p_d1	[0.2, 0.8]
		p_d2	[0.5, 1.5]
15	3.6	p_d3	great than 0
		p_l	[5, 15]
		p_ac	[0.6, 1]
		p_c3	[0.6, 1]
		p_k	[1, 10]
20	3.8	p_d4	[10, 100]
		p_c4	[0.6, 1]
	3.9	p_ag	[0, 7]
		p_ga	[0.01, 0.1]
		p_nm	[8, 20]
25	3.10	p_l2	[5, 20]
		p_r1	[1, 10]
		p_r2	[10, 20]

The invention thus provides a method for calculating the  
 25 global difference between two stripe patterns by means of  
 their global features used in finer classification and  
 search.

While the description above refers to particular  
 embodiments of the present invention, it will be understood  
 30 that many modifications may be made without departing from  
 the spirit thereof. The accompanying claims are intended to  
 cover such modifications as would fall within the true scope  
 and spirit of the present invention.

The presently disclosed embodiments are therefore to be  
 35 considered in all respects as illustrative and not  
 restrictive, the scope of the invention being indicated by  
 the appended claims, rather than the foregoing description,

WO 93/07584

PCT/US92/08446

49

and all changes which come within the meaning and range of  
equivalency of the claims are therefore intended to be  
embraced therein.

WO 93/07584

PCT/US92/08446

50

## WHAT IS CLAIMED:

1. An automatic method for identifying an image having varying brightness values constituting a pattern of stripes containing minutiae, comprising:
  - dividing the image into a rectangular matrix of image points and providing a representation of the gray level value of the brightness at each image point;
  - transforming the representations into a selected brightness range;
  - calculating the average direction of local texture at each point on the image with a function of gradient models in a neighborhood of the point;
  - calculating the curvature or the inconsistency of directions of local texture at each point on the image with a function of gradient models in a neighborhood of the point;
  - separating a useful portion of the image, which contains an accurate representation of the pattern of stripes, from noisy background;
  - extracting the global features of the image to represent the direction array by selecting some points on the image and storing the directions at these points;
  - finding singularities in the useful portion by comparing curvature values of selected matrix points in the useful portion;
  - producing representations of the shapes of selected stripes of the image based on the curvature and direction values;
  - locating a standardized coordinate axis system having an origin on the image by locating an intersection of lines normal to selected stripes in a selected region of the image;
  - producing representations of selected characteristics of the shape of one of the selected stripes on the basis of the distance between the origin of the coordinate axis system and points on the selected stripe in selected directions from the origin and producing a

WO 93/07584

PCT/US92/08446

51

representation of a global feature function of the  
fingerprint based on a set of values each representing a  
difference between the direction between the origin of the  
coordinate axis and a point on a stripe and the direction of  
5 the stripe pattern at that point,  
determining the locations of minutiae in the image  
on the coordinate axis system from the transformed  
representations of the gray level values of the brightness at  
each image point; and  
10 producing a representation of the quality level of  
the image.

2. The method defined in claim 1 wherein the image  
represents a fingerprint.

15

3. The method defined in claim 2 comprising employing  
a quick recurrent algorithm for calculating the direction and  
curvature with same size as the image, by the steps of:

(1) Calculating four gradient model arrays about 0, 45,  
20 90 and 135 degrees respectively;

(2) Calculating the average gradient models in each  
neighborhood recurrently;

(3) Using tables instead of operations of arc-tangent,  
wherein the curvature is a measurement of accuracy about the  
25 direction at the same point.

4. The method defined in claim 2 comprising segmenting  
an octagonal clear region of fingerprint ridges from  
background and noise by means of eight straight lines  
30 according to the curvature array.

WO 93/07584

PCT/US92/08446

52

5. The method defined in claim 2 comprising analyzing the ridge trends around a point in image by means of the difference of local direction and vector direction at each point on the digital circles with various radius and deriving a Fourier transform and reverse transform for deciding the forkedness by power spectrum and finding the trends by filtered differences.

6. The method defined in claim 2 comprising performing line tracing based on direction and curvature values by accumulating errors of coordinates and directions to correct the trace for extracting contour lines, shape lines and normal lines.

7. The method defined in claim 2 wherein the coordinate axis is located by locating the center and central orientation of the fingerprint macroscopically by means of a vault line and normal lines.

8. The method defined in claim 2 comprising locating the 'a' core of a plain arch and its main trend macroscopically that are consistent with other types of fingerprints.

9. The method defined in claim 2 wherein the shape features for both loops and whorls are extracted from shape lines, the shape features of a whorl being composed by two parts of two loops referring to two deltas respectively.

10. The method defined in claim 2 comprising producing a general classification by means of relations among shape lines and fine classification by means of shape features.

WO 93/07584

PCT/US92/08446

53

11. The method defined in claim 2 comprising identifying global features by angle values on some points selecting from the direction array, the points are located spatially on a lattice or on several circles with a common center.

12. The method defined in claim 2 wherein the minutiae are located in terms of endings or bifurcations of valleys.

13. The method defined in claim 12 wherein valleys are traced and minutiae located on the gray level image by means of contour tracing referring to gray levels, comprising the steps of:

- (1) selecting starting point for valley tracing,
- (2) connecting two terminals of traces which satisfy certain conditions,
- (3) changing the gap of valleys.

14. The method defined in claim 2 wherein the quality level of the fingerprint is determined on the basis of quality vectors referring to the position of center, number of minutiae, noise level, and area of clear region in order to decide automatically or suggest the operator for acceptance, rejection or reinput of the fingerprint image.

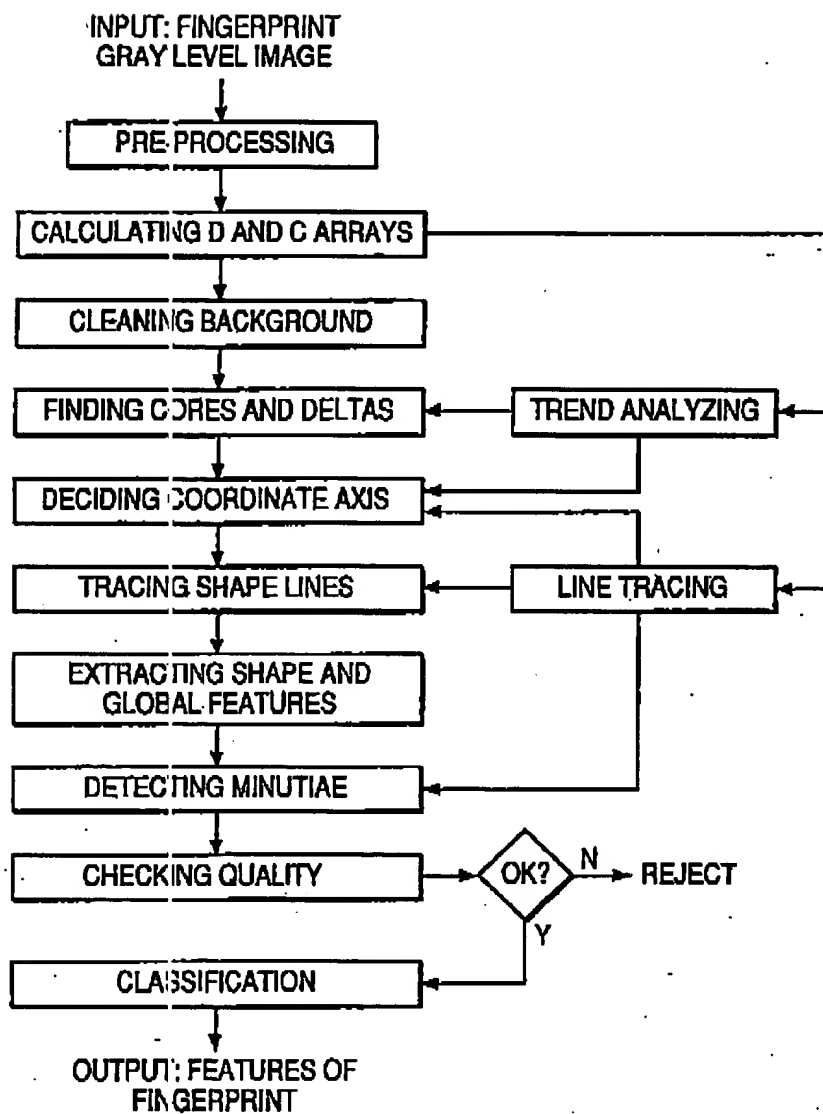
15. The method defined in claim 2 further comprising calculating global difference between two fingerprints by means of their global features used in finer classification and search.

WO 93/07584

PCT/US92/08446

1/9

FIG. 1

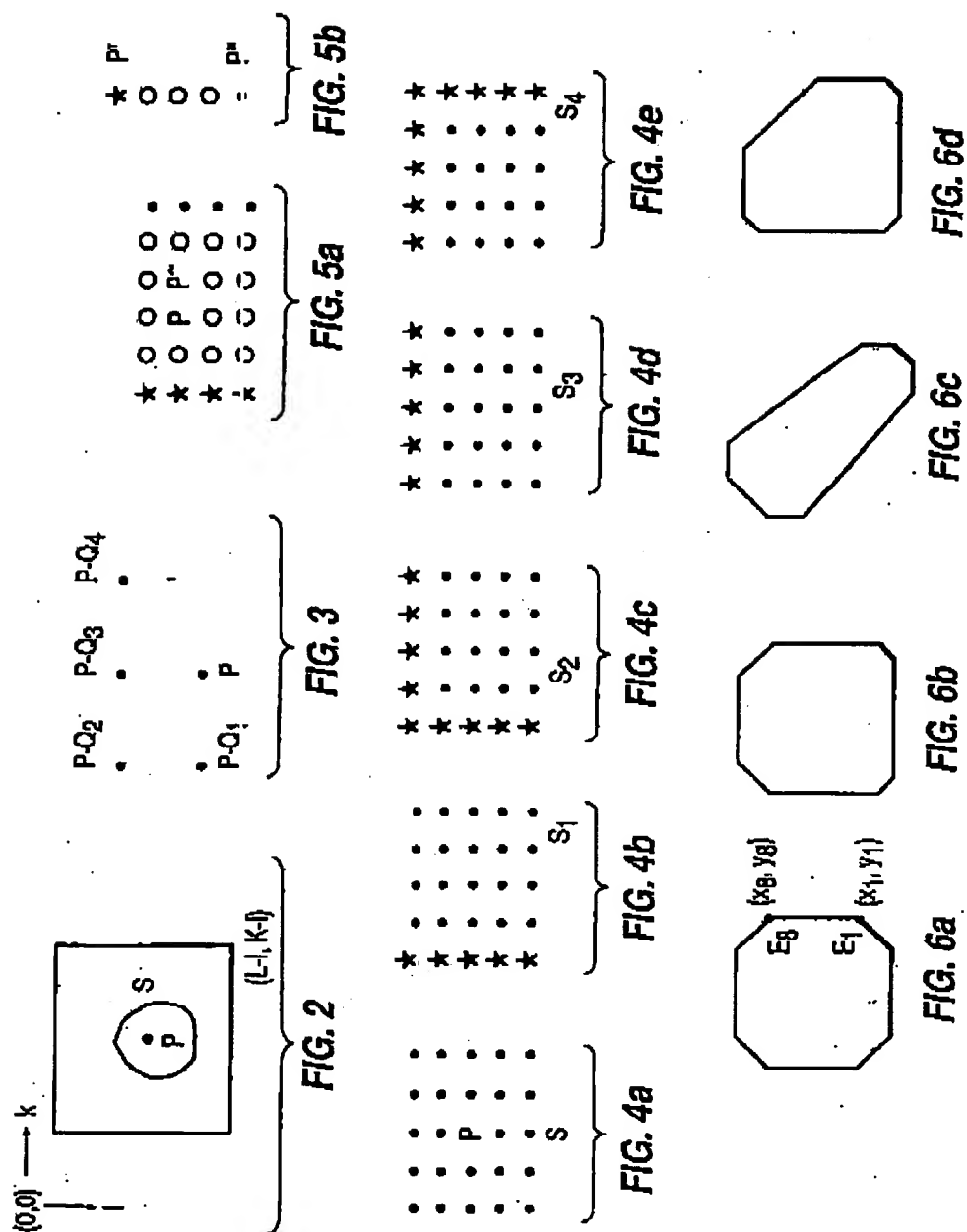


SUBSTITUTE SHEET

WO 93/07584

PCT/US92/08446

2/9



WO 93/07584

PCT/US92/08446

3/9



FIG. 7e

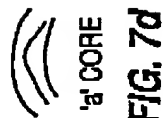


FIG. 7d

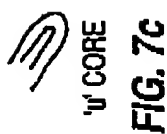


FIG. 7c



FIG. 7b



FIG. 7a



FIG. 8e

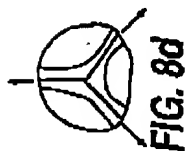


FIG. 8d



FIG. 8c



FIG. 8b

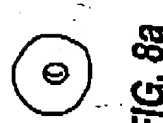


FIG. 8a

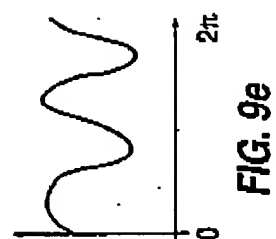


FIG. 9e

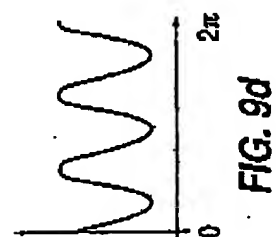


FIG. 9d

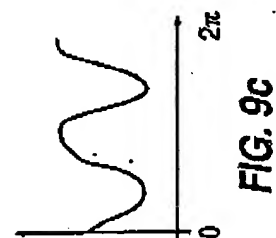


FIG. 9c

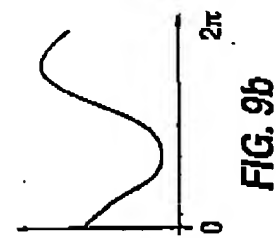


FIG. 9b

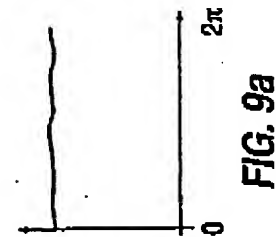
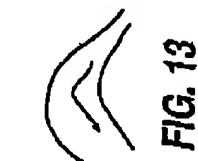
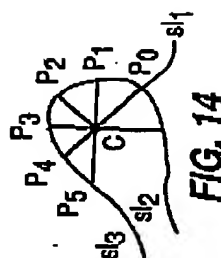
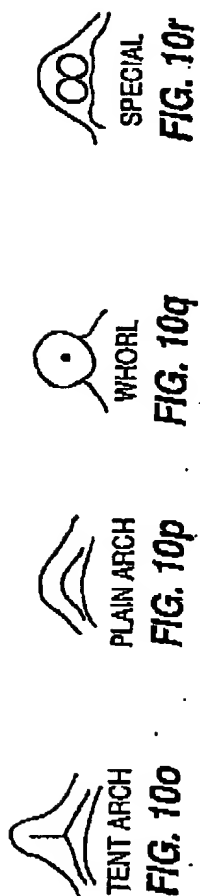
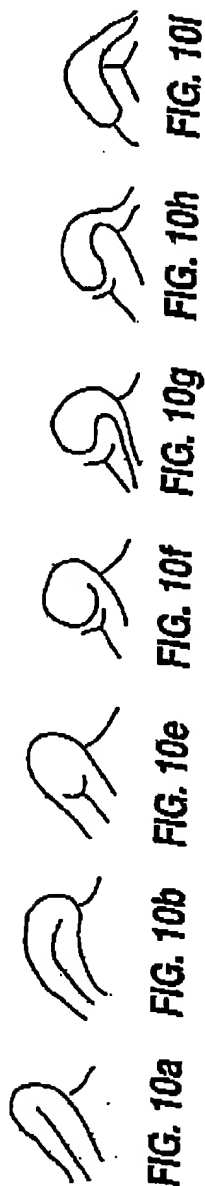


FIG. 9a

WO 93/07584

PCT/US92/08446

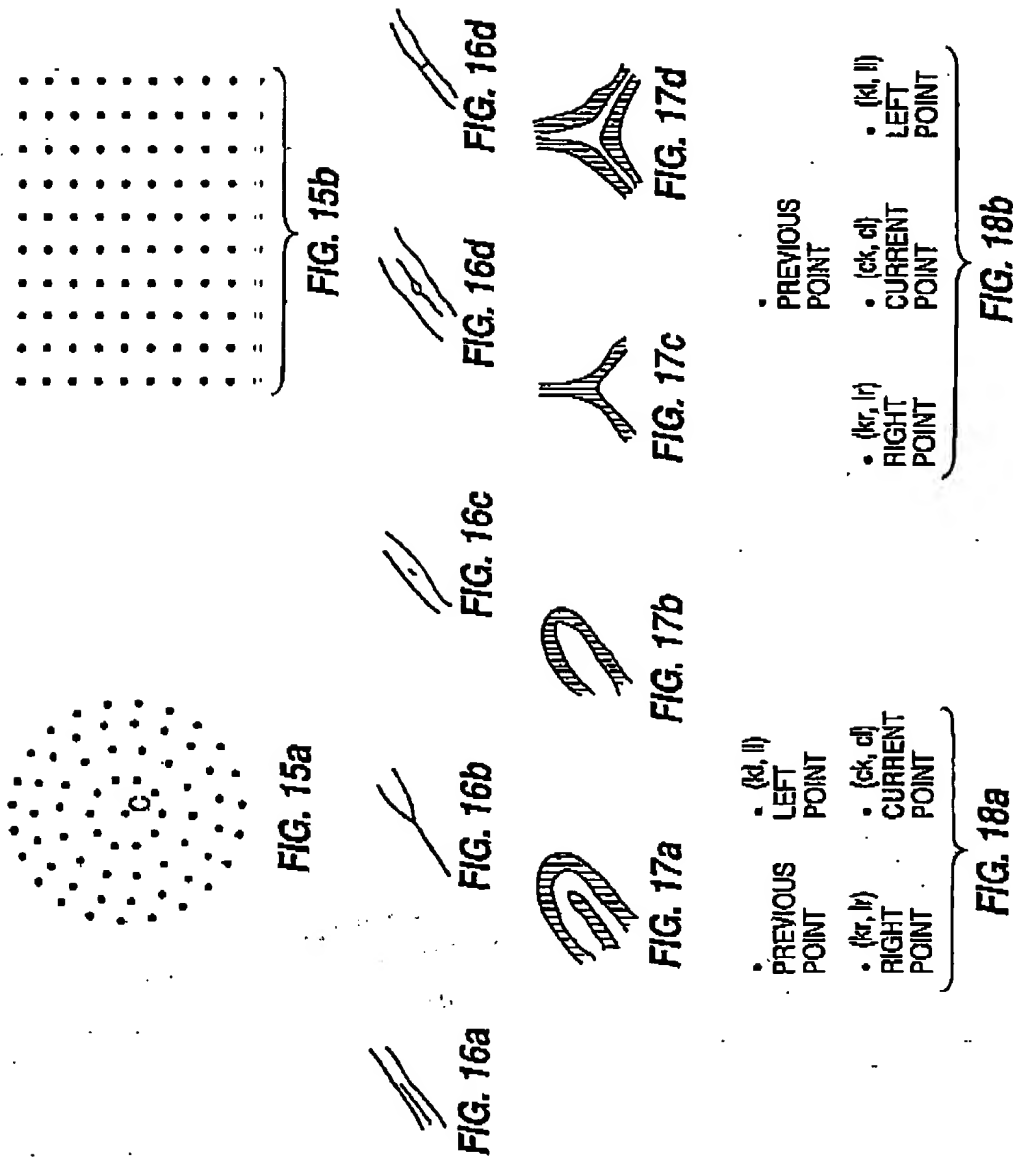
4/9



WO 93/07584

PCT/US92/08446

5/9



WO 93/07584

PCT/US92/08446

6/9

FIG. 19

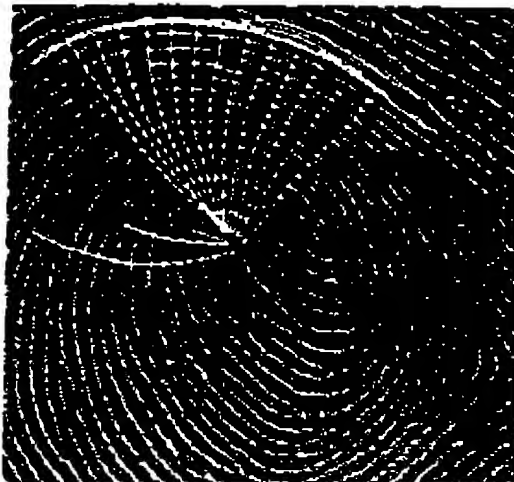
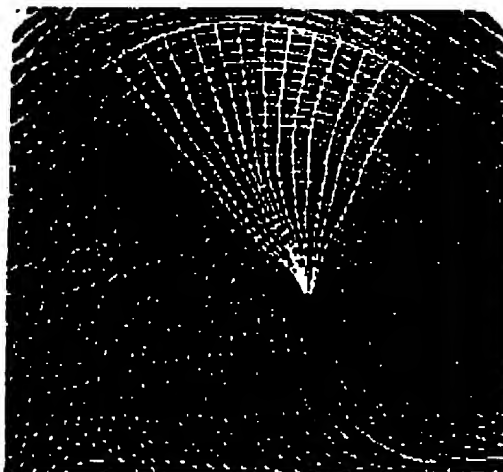


FIG. 20



QUARTITE SHEET

WO 93/07584

PCT/US92/08446

7/9

FIG. 21

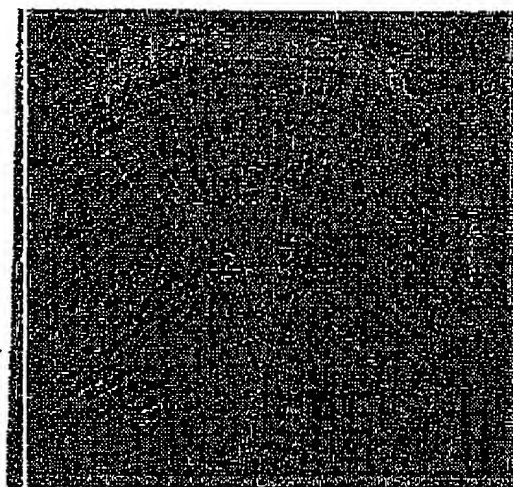
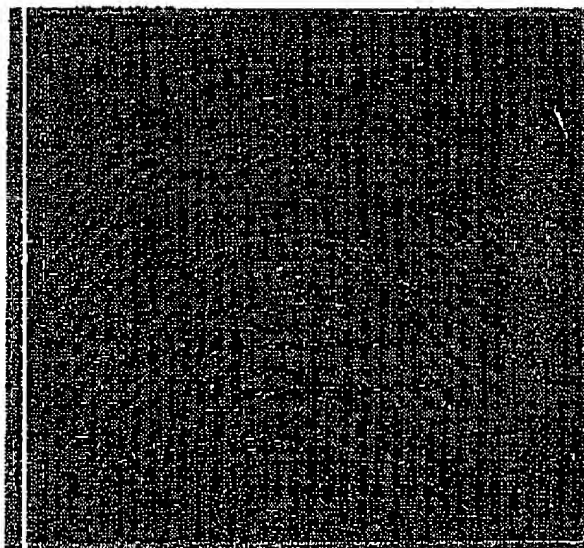


FIG. 22



CONTINUITY SHEET

WO 93/07584

PCT/US92/08446

8/9

FIG. 23

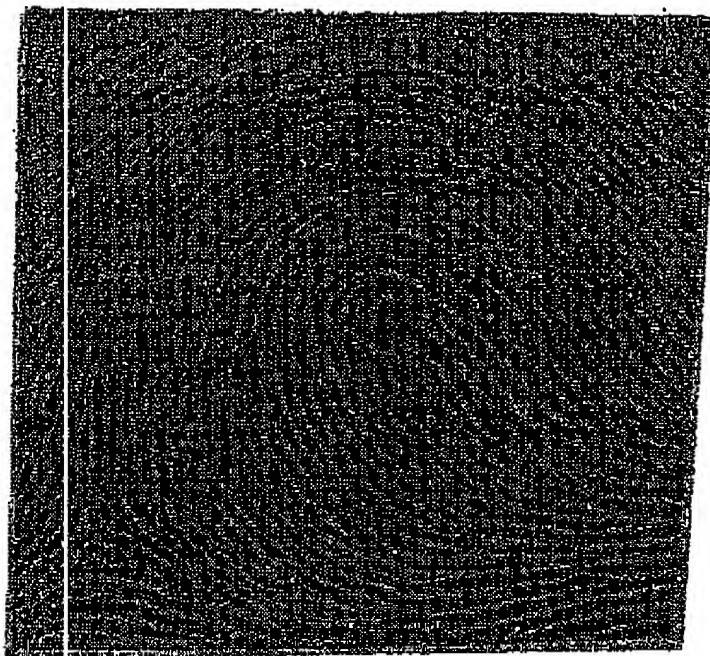
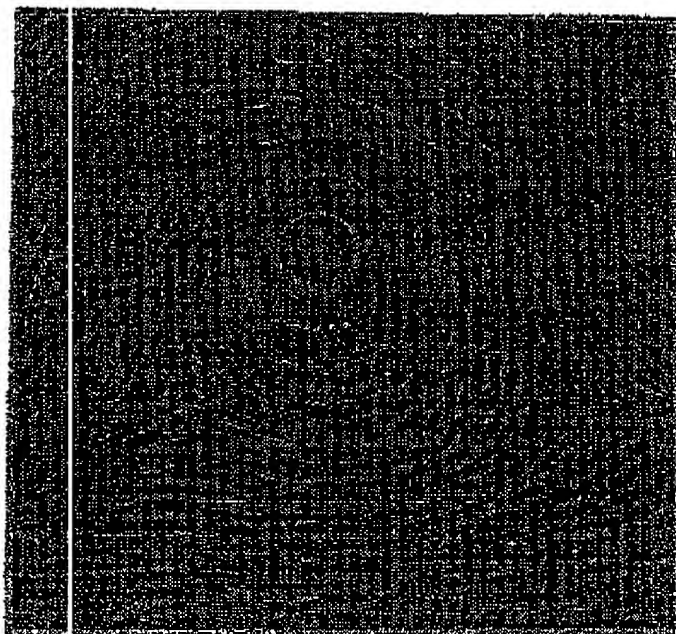


FIG. 24

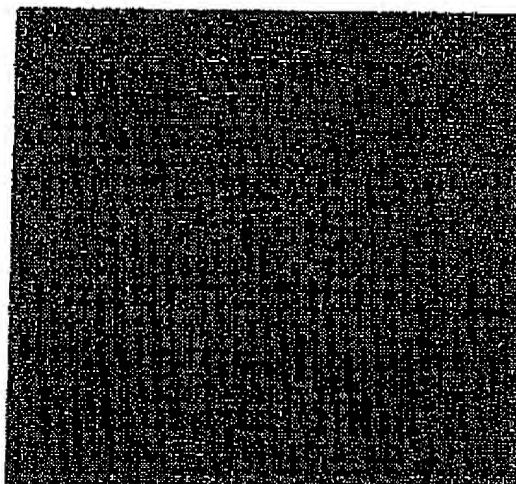


WO 93/07584

PCI/US92/08446

9/9

FIG. 25



SUBSTITUTE SHEET

## INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/US92/08446

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) : G06K 9/00

US CL : 382/5

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 356/71; 382/4

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US,A, 4,083,035 (RIGANATI ET AL) 04 APRIL 1978 See entire document	1-15
A,P	US,A, 5,067,162 (DRISCOLL, JR ET AL) 19 NOVEMBER 1991 See entire document.	1-15
A,P	US,A, 5,140,642 (HSU ET AL) 18 AUGUST 1992 See entire document.	1-5

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:	T	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
*A* document defining the general state of the art which is not considered to be part of particular relevance	X	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
*B* earlier document published on or after the international filing date	Y	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
*L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	A*	document member of the same patent family
*O* document referring to an oral disclosure, use, exhibition or other means		
*P* document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

11 DECEMBER 1992

Date of mailing of the international search report

05 JAN 1993

Name and mailing address of the ISA/W,  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Authorized officer

/s/ LEO H. BOUDREAU

INTERNATIONAL DIVISION

Facsimile No. NOT APPLICABLE

Telephone No. (703) 305-4762

Form PCT/ISA/210 (second sheet) (July 1992)\*

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**